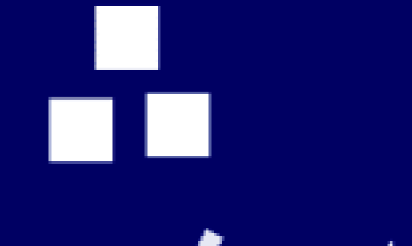
JAVASCRIPT



الجافا سكربت



teach yourself

إعداد :

م. سامي الربايعة



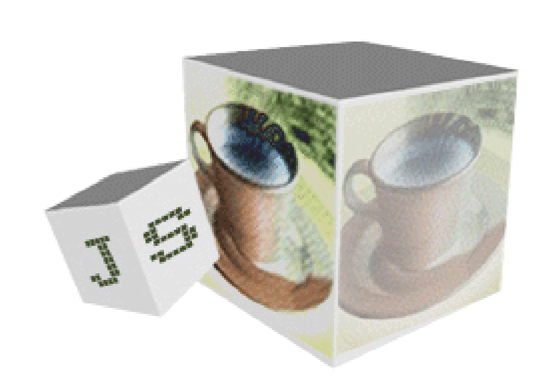
نسخة مجانية

تعليم الجافا سكربت
JAVASCRIPT LERNNING

إعداد :
م . سامـــي الرבايعـــة

By :
Eng . SAMI AL-RABA'AH

Email :
e_book@myway.com



المملكـة الأردنيـة الهاشميـة

Feb 2004

الفهـــرس

- مقدمة عامه عن لغة الجافا سكربت :
§ JAVASCRIPT :

الجافا سكربت هي لغة برمجه اذا جاز التعبير تلعب دور حيوي وفعال في صفحات الويب من خلال القيام بوظائف قد تكون خارجية او داخلية بل لنكن اكثر دقة هي مرنه الى درجه تجعلك تتحكم بكل جزء من اجزاء صفحة الويب كأن تستخدمها في ال forms اي النماذج او كنوافذ تخرج للمستخدم لتخبره بامر معين او تنبه وغيرها من الاستخدامات التي اراها من وجهة نظري هي الروح التي يبثها مطور الويب في صفحاته اثناء برمجته .

اما من اين جاءت لغة الجافا سكربت فهي جاءت من خلال اهتمام شركة Netscape لتضفي الحيوية الى صفحات الويب ولا يفوت على بال احد ان شركة Sun هي الاساس والام في لغة الجافا بشكل عام فما بالك بالجافا سكربت فلغة جافا سكربت هي لغة بسيطه نوعا ما وسهله التبويب والتحكم بل يستطيع البرمجه بها من لم يتعلم اي لغة في حياته وهذا ما ركزت به في كتابي هذا بان اجعل المتدرب يبدأ من الصفر فليس هدفي فهم لغة الجافا سكربت فقط فحسب بل التكنيك البرمجي بإختلافاته فالمبدأ واحد على اختلاف شيفرة الكود فال for loop مثلا او ال if condition او غيرها موجوده في جميع لغات البرمجه ولهذا اعتبر كتابي هذا باب لدخول منه الى فهم اللغات القادمه التي سوف اطرح بها كتب بإذن الله وكذلك الى لغات البرمجه الموجود كتبا عنها في الاسواق او النت فهذا الكتاب بدايتك لفهم بقية اللغات بكل سهوله او على الاقل بيسر ودون عناء اذا ليس كل اللغات ليكن بعضها واذا ليس بعضها اذا على الاقل جزء من كل لغة ...

عزيزي المتدرب لغة الجافا سكربت بإعتقادي الخاص كمطور ويب مهمة نوعا ما فانت عندما تصمم في HTML انت تصمم واجهة ولكنها خالية من الحركة بل انك لا تستطيع ان تغير شيء منها إلا بالتغير الكود نفسه ولكن تخيل مثلا انك تريد ان تظهر بنر اعلاني يتغير كل وقت معين من غير ان تلمسه بيدك اي من غير ان تغير الكود بل هو من نفسه يجلب بنر الاعلان الاول ثم بعد دقيقة مثلا يغيره الى بنر اعلان ثاني وهكذا او تصور انك تريد ان تضع في رأس صفحتك التاريخ والوقت او كأن تريد ان يظهر اسم زائرك في صفحتك لكي تشده لها وغيرها من الافكار فالجافا سكربت تفعل لك كل هذا وبكل بساطه وسهولة ..

اما في البرامج او الاكواد التي سوف نتعلم كتابتها هي مجموعة من الأوامر التي تكتب مجتمعة أو متفرقة ليعمل البرنامج بصورتة الرئيسيه ولنقل هي الاساس او البنية التحتية لبرنامجنا ولا تقلق تستطيع ان تكتب برامجك او اكوادك من خلال برنامج ال NOTEBAD المتوفر في كل جهاز وإنني افضل ان تجد محرر للجافا سكربت في الانترنت وهذا افضل لكي تعرف موقع الخطأ مباشرة من غير عناء .

وفي نهاية المقدمه هذي اريد التنبيه الى شيء اراه كثيرا في الانترنت وهو يوجد فرق كبير عندما نقول هذا كود JAVA او كود JAVASCRIPT فالاختلاف كبير بينهم وإن كان المسمى يوحي بغير ذلك وايضا يوجد فرق بسيط بين ال DHTML والجافا سكربت وإن كنا نستخدم في DHTM اوامر واكواد ال JAVA SCRIPT بل يكون الاعتماد الكبير عليها ولكن هذي عزيز المتدرب لغة اذا جاز تسميتها ثانيه ولها تطبيقاتها الحركية والتأثيرية الخاصه وإني ارى اكثر المواقع تعرض للمستخدمين اكواد جاهزه للتحميل او النسخ وتكتب عنها انها JAVASCRIPT وإني ارى كثيرا منها تسميته الصحيحه DHTML اي الداينمك وليس الجافا سكربت وانا اقول كثيرا منها وليس جميعها .. ونحن هنا لسنا في محض المقارنه ال JAVASCRIPT وال JAVA و DHTML فلكلا منها اكواده الخاصه واسلوبه وتكنيكه ويجب هنا التنبيه ان ال JAVA بعيدة كل البعد عن ال JAVASCRIPT على عكس ال DHTML القريبة جدا فلا اقارن بين ال JAVA و ال DHTML مع ال JAVASCRIPT وكانني اساويهم ببعض فال DHTML قريبة جدا او تصب في مصب نفسه ولكن ال JAVA بعيدة كل البعد بل لا يوجد مقارنه ولها اعمالها واكوادها الخاصه وعملها الخاص .

ويوجد بعض الملاحظات المهمه قبل البدء بتعلم JAVASCRIPT ارغب التنبيه لها ولكي لا اذكرها في متصف الكتاب وتتشتت افكارك .

§ قمت في هذا الكتاب بالتدرج على اسلوبي الخاص في تعليم المتدرب هذه اللغة وهذا اجتهاد مني لهدفين هو شد المتدرب على الاستمرار في التعلم واخذ الفائده التي يرجوها من اول صفحة يقرأها ولهذا كان لي اجتهادي الخاص في التدرج في صب المعلومه وقد استخدمت عبارة اذا جاز التعبير كثيرا وهنا اكون قد تجاوزت في عبارة معينه في موصفها عن وصفها البرمجي البحث وهذا لغرض إيصال المعلومه وهذا الذي يهمني كثيرا.. لذلك استخدم هذه العبارة في اماكن معينه .

§ سوف نتطرق في نهاية الكتاب الى الكائنات اي ال Object وسوف اكتب لك مقدمة عنها هنا لكي تكون على علم بها وما هيتها . ال .Object كائن يحتوي علي خواص ، ومنه طرق تتيح لك استخدام تلك الخواص علي بيانات محددة ومعينه مما يعطي قوة كبيرة نوعا ما للجافا سكربت بل هي قائمه على ذلك وما ينعكس عليها اي ال JAVASCRIPT من بساطة تصميم برامجها وسهولة وسلاسة تطويرها يشار أحيانا للغات

التي تتيح استخدام الكائنات بكامل خصائصها علي إنها لغات برمجة موجها للكائنات OOP مثل الجافا و السي ++ الى غيرها من لغات البرمجه ولا اريد ان اخوض بتاريخها وادخل بك الى صفحات عده وأرهقك في القراءة وانت لم تبدأ بعد ولكن انصحك بالقراءة عنها في الانترنت فكل شيء يعتبر مرجع مفيد لك حتى لو كانت معلومة واحده .

§ الكلمات المحجوزة وسوف اذكرها لك فكما في جميع اللغات يوجد هناك كلمات محجوزة اي Keywords وسميت كذلك لكي لا تستخدم إلا بمواقع خاصه في كود اللغة وليس كمسميات تستخدمها .
وهي كلمات خاصه تقوم بعمليات ووظائف معينه على متغيرات اسندت او مررت لها اذا جاز التعبير لذلك مثلا لا يجوز استخدام ايا منها في اسم دالة على سبيل المثال او غيرها بل ابتعد في تمساتك الخاصه عن هذه الكلمات المحجوزة اذا ما هي الكلمات الحجوزة في الجافا سكربت ؟ هي :

breake	**case**	**continue**	**delete**	**do**
else	**false**	**for**	**function**	**if**
in	**new**	**null**	**return**	**switch**
this	**true**	**typeof**	**var**	**voide**
while	**with**			

لا تتزعج من عدم فهمك بعض الملاحظات فهذا امر طبيعي ولكنني اذكرها فقط لك لتعلم بها وفي اثناء قراءتك الكتاب سوف تعرف ما قصدته في هذه الملاحظات .

اعذروني اذا تجاوزت في شرحي في بعض العبارات في اثناء شرحي في الكتاب فهدفي هو إيصال المعلومه وليس عرض العضلات البرمجيه والمصطلحات المعقده وخصوصا انني لست من الناس الذين يفضلون التعريب في مواقع معينه وذلك لان عند تعريب مصطلحات معينه تفقد معناها الصحيح مع اعتزازنا في لغتنا العربية اللغة الأم لذلك كنت اعربها احياننا ليس حرفيا بل حسب ما يمكن ان تصلك المعلومه وهذا ما يهمني فكثيرا ما نقرأ كتب رائعه تحمل مصطلحات عده وزخم برمجي اصطلاحي وفلسفي ولكن ماذا نخرج بالاخر لا شيء وما اكثر الكتب في ذلك لذلك اذا البساطه خير سبيل وإن كان هناك من يعترض عليها .

§ لم اضع لهذا الكتاب فهرس مفصل لان من وجهة نظري ان الكتب التي اصدرها هي بمثابة دورات تدريبية تعتمد اسلوب المخاطبه والاتصال بالمتدرب والتدرج به حسب تفكيره والإجابة على كل ما يجول في خاطره قد يكون هناك احد له وجهة نظر اخرى ولكن لكلا منا اسلوبه الخاص به كما انني اعتمدت على جعل بعض الأمثلة التي تحتوي على ارقام تبدأ من اليسار الى اليمين تستطيع قلب ذلك بوضع dir = rtl في وسم ال html ببداية الصفحة .

§ **ملاحظة مهمة :** ارجو عدم نسخ الكود في الكتاب ووضعه في ال Notepad مباشرة فقد لا ينفذ معك وذلك لانني احياننا اترك فراغا بين فاصله او نقطه او كلمة او قوس او ما شابه ذلك وهذا بسبب انني اكتب الكود في برنامج ال word فلا يبدو الفراغ واضح فأضطر الى جعله فراغين او اكثر لكي تراه انت انه فراغ واحد ومن هنا كل البرامج الموجوده التي كتبت في الكتاب مرفقه مع الكتاب كصفحات انترنت تستطيع استخدامها بدلا من النسخ مباشرة واللصق .

§ **طرق كتابة كود الجافا سكربت ..**

١- الطريقة الأولى :

```
<script type = " text/javascript">
                  -
                  -
                  -
         هنا يكتب كود الجافا سكربت
                  -
                  -
</script>
```

٢- الطريقة الثانية :

```
<script language="JavaScript">
               -
               -
               -
        هنا يكتب كود الجافا سكربت
               -
               -
</script>
```

يجدربنا المعرفة هنا ان كود الجافا سكربت تستطيع كتابته ببرنامج **(notepad)** او بأي محرر اخر مختص بكتابة كود الجافا سكربت .

§ **موضع كود الجافا سكربت ..**

يكتب عادة كود الجافا سكربت في رأس صفحة **(html)** اي بين وسمي رأس الصفحة :

(<head> ----------- </head>)

ملاحظة :

قد يضع البعض كود الجافا سكربت في ذيل الصفحة او في جسمها وهذا وارد الحدوث ..

§ **تمثيله :**

```
<html>
    <head><title>الجافا سكربت</title>
      <script type = " text/javascript">
                -
         يكتب هنا الكود المرغوب به
                -
      </script>
    </head>
 <body></body>

</html>
```

تعريق مبدأي ببعض اوامر الجافا سكربت ..

١- أمر الطباعة :

الطريقة الأولى : طباعة جملة لا يقع عليها تأثير اي وسم من وسوم الـ **(html)** ...

```
document.write("هنا الجملة المراد طباعتها ");
```

الطريقة الثانية : طباعة جملة يقع عليها تأثير وسم ال **(html)** ...

```
document.write("<h1> هنا الجملة المراد طباعتها </h1>");
```

الطريقة الثالثة : طباعة جملة يقع عليها تأثير وسم ال **(htm)** و ال **(style)** ...

```
document.write("<h1 style=\"color : red\"> هنا الجملة المراد طباعتها </h1>");
```

الطريقة الرابعة : طباعة قيمة متغير ...

```
document.write( sum );
```

الطريقة الخامسة : طباعة قيمة متغير يسبقه جملة نصية **..**

```
document.write(": الناتج هو  " +  sum );
```

الطريقة السادسة : طباعة قيمة متغير يسبقه ويعقبه جملة نصية ..

```
document.write(": الناتج هو  " +  sum + " للعملية ");
```

ملاحظات مهمة :

- اي جملة نصية تريد ان تضعها بداخل كود الطباعة يجب ان تحصرها بين **(")** حتى لو احتوت على وسم ال **(html)**

- طباعة قيمة متغير .... ويوجد لها اكثر من حالة :

١- اذا كنت تريد فقط ان تطبع قيمة المتغير بمفرده ... تضعه بداخل أمر الطباعة من غير ان تحصره بـ **(").**
٢- اذا كنت تريد طباعة قيمة متغير تسبقه جملة نصية يجب ان تحصر الجملة النصية بين **(")** ومن ثم تذكر المتغير المراد طباعة قيمته ولكن يجب ان تضع قبله **(+ ")** ولتوضيح اكثر تضع اشارة **(+)** بالجهة التي بها النص....

مثلا ... جاء النص قبل المتغير نضع اشارة الـ **(+)** قبل المتغير

```
document.write(" : الناتج هو  " +  sum );
```

مثلا ... جاء النص بعد المتغير نضع اشارة الـ **(+)** بعد المتغير

```
document.write( sum +" : الناتج هو  "  );
```

مثلا ... جاء النص بعد المتغير وقبله نضع اشارة ال **(+)** بعد المتغير وقبله ..

```
document.write(": الناتج هو  " +  sum + " للعملية ");
```

يعني بإختصار ضع اشارة **(+)** بجانب المتغير المراد طباعته من الجهة التي بها الجملة النصية واذا لم يكن هناك جملة نصية اذا لا تضع اشارة **(+)**

سوف نذكر ما هو المتغير وما نقصد به لاحقا فلا تقلق فما عليك سوى معرفة طباعته اما ما هو ولأي غرض يستخدم سوف تعرف لاحقا ..

- o تستطيع ان تستخدم **(document.writeln)** بدلا من **(document.write)** فبكتابة الأول يترك مسافة بين كل كلمة في جملة الطباعة اذا "جاز التعبير" بشكل عملي ..
- o لطباعة جملة في سطرين اي " النزول الى سطر جديد " نضع بداخل جملة الطباعة **(
)** قبل الجملة التي تريد ان نجعلها في سطر جديد ..

مثل :

```
document.write(" Arab <br />Top ");
```

الطباعة سوف تكون بهذا الشكل :

Arab
Top

اي كل كلمة في سطر ...

٢- خروج نافذة للمستخدم نحدد نحن ما يكتب بها وهي خاصه في كائنات النوافذ Window Object.

كهذه النافذة :



وتسمى هذه النافذة بالجافا سكربت **(alert)** ويكتب كودها بهذه الطريقة ...

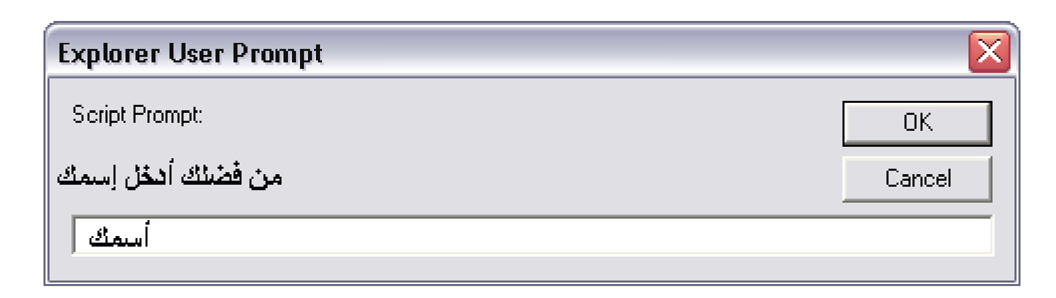
```
window.alert(" اهلا بك فى موقعنا ");
```

*** بعض الاوامر التي تستخدم داخل كود ال (alert):**

الامر	وصف الأمر	مثال ..
\n	سطر جديد	window.alert("hi \n all");
\t	لترك مسافة بين كل كلمة وكلمة بحيث تظهر كجدول	window.alert("hi \t all");
\r	لوضع كل كلمة بسطر ولكن بإختلاف ال position	window.alert("hi \r all");
\\	لكتابة رمز (\ )	window.alert("hi \\ all");
\"	لكتابة رمز (")	window.alert("hi \" all");
\'	لكتابة رمز (')	window.alert("hi \' all");

٣- خروج نافذة للمستخدم يحدد هو ما يكتب بها " اي نافذه قراءة من لوحة المفاتيح " وهي خاصه بكائنات النوافذ Window Object .

كهذه النافذة :



وتسمى هذه النافذة بالجافا سكربت **(prompt)** ويكتب كودها بهذه الطريقة ...

```
window.prompt ("من فضلك أدخل إسمك","إسمك");
```

ملاحظة :
يأخذ كود الـ **(prompt)** بداخله قيمتين الاولى يكتب بها ما سوف يخرج للمستخدم في أعلى النافذة .. والقيمة الثانية يكتب بها قيمة ابتدائية بداخل صندوق الكتابة وهذه القيمة ليس لها اي قيمة فعلية فهي ما تلبث حتى ينتهي دورها بمجرد ان يكتب المستخدم مكانها اسمه او اي شيء يريده ... جرب ان تضع هذا الامر وإنظر ماذا يخرج لك والذي يعتبر ايضا من كائنات النوافذ **Window Object**

```
var con = confirm("هل تريد حفظ البيانات ؟");
```

وهي عبارة عن نافذة تخرج للمستخدم لسؤاله عن قبول أو رفض شيء معين أنت تحدده .

§ **المتغيرات :**
نظرة سريعة على المتغيرات :
كما تعلم في أي لغة برمجة تعتبر المتغيرات ذو فائدة كبيرة فبواسطتها نستطيع التحكم ببرنامجنا بكل سهولة ...
ففي لغة الجافا سكربت تعرف المتغيرات بـ **(var)**.

مثال :

```
var name  ;
```

ويجدر التنبيه هنا انه أذا اردنا أن يكون المتغير قيمة رقمية **(integer Or float)** " أعداد صحيحة أو أعداد ذات أرقام عشرية " فإننا نقوم بتحويلها بهذه الطريقة :

1- parseInt (أسم المتغير هنا) ;

وهذا سوف يحول العدد الى عدد صحيح **(integer)** أي يكون كتابة الكود كالتالي :

```
var number = 55 ;
parseInt (number) ;
```

2- parseFloat (أسم المتغير هنا) ;

و هذا سوف يحول العدد الى عدد عشري **(float)** أي يكون كتابة الكود كالتالي :

```
var number = 5.4 ;
parseFloat (number) ;
```

ملاحظة :

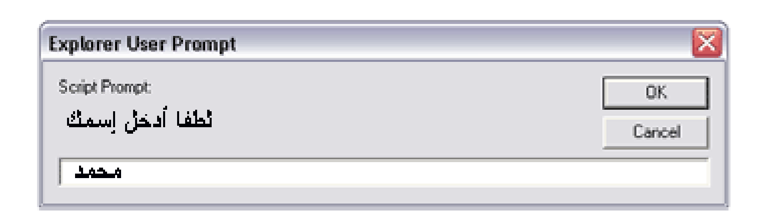
عليك الإلتزام بشكل الحرف أي الحرف الصغير والكبير **(parseFloat)** .
..وللحديث بقية سوف يذكر في وقته.

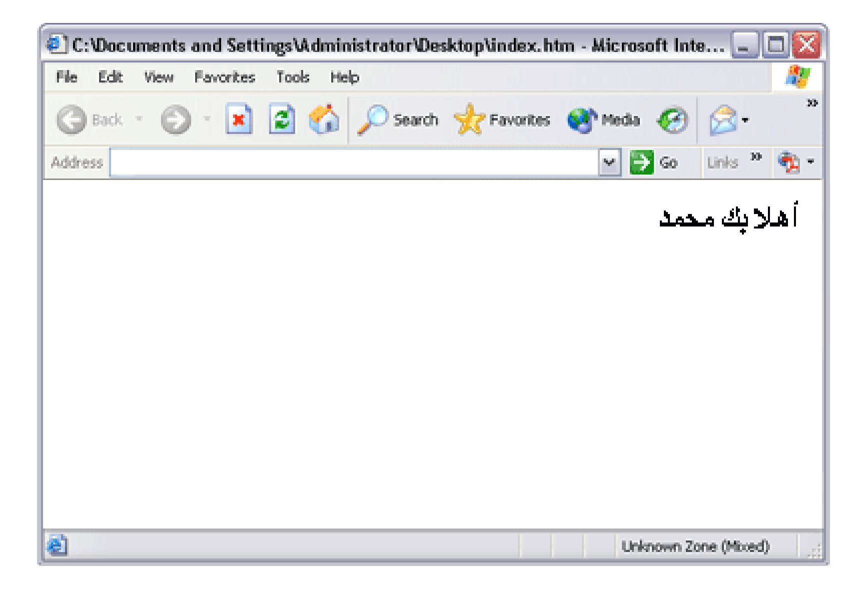
لنأخذ أمثلة على ما سبق ونطبقها ...

مثال ١ :

```
<html dir="rtl">
  <head><title>الجافا سكربت</title>
    <script>
      var name ;
      name = window.prompt ( "لطفا أدخل إسمك " , "إسمك" ) ;
      document.write( " أهلا بك " + name  ) ;
    </script>
   </head>
 <body> </body>

</html>
```





في هذا المثال قمنا بما يلي :
عرفنا متغير وجعلنا اسم المتغير هو **(name)** .

جعلنا هذا المتغير يساوي القيمة التي سوف يدخلها المستخدم من لوحة المفاتيح وذلك من خلال الأمر **(window.prompt)** . فبمجرد خروج النافذة للمستخدم ويكتب بها اسمه سوف يخزن الأسم المدخل بداخل المتغير **(name)** وهذا واضح من العبارة :

```
name = window.prompt ( "لطفا أدخل إسمك ", "إسمك" ) ;
```

ثم قمنا بطباعة المتغير **(name)** وهذا من خلال الأمر :

```
document.write( " أهلا بك " + name  ) ;
```

وبالتالي سوف يطبع لنا ما خزن داخل المتغير وهو إسم المستخدم الذي أدخله . وبعد ذلك سوف يطبع الجملة النصية التي كتبناها وهي " أهلا بك " ... أي سوف يطبع لنا الجملة التالية على فرض اننا قمنا بإدخال الإسم " محمد " : **" أهلا بك محمد "**

مثال ٢ :

```
<html>
    <head><title>الجافا سكريبت</title>
      <script type = "text/javascript" >
         window.alert ("أهلا بك") ;
      </script>
    </head>
  <body></body>

</html>
```



وفي هذا المثال خروج نافذه للمستخدم وهي نافذة (alert) التي ذكرناها سابقا ..

§ العمليات الحسابية :

أسم العملية الحسابية	الصيغة	خوارزمية الصيغة	مثال
عملية الجمع	+	x + y	x + y
عملية الطرح	-	y –x	x – y
عملية الضرب	*	x * y	X * y
عملية القسمة	/	x / y	x / y
عملية باقي القسمة	%	x % y	x % y

§ العمليات المنطقية :

الصيغة	تمثيلها بالجافا	وصف الصيغة	مثال
=	==	x == y	قيمة x تساوي قيمة y
≠	!=	x != y	قيمة x لا تساوي قيمة y
>	>	x > y	قيمة x أكبر من قيمة y
<	<	x < y	قيمة x أقل من قيمة y
>=	>=	x >= y	قيمة x أكبر أو تساوي قيمة y
<=	<=	x <= y	قيمة x أقل أو تساوي قيمة y

ملاحظة :

يوجد أولويات تعتمد في العمليات الحسابية البرمجية.

- **لنأخذ بعض العمليات الحسابية بشكل سريع :**

```
var number = 4 ;
number+ = 2 ;
```

المتغير **(number)** كان يحمل القيمة **(4)** ثم أضفنا له **(2)** فأصبح الناتج **(6)**.

ويمكن أن نكتب العمليات السابقة بطريقة برمجية أخرى وسوف يخرج لنا نفس الناتج :

```
var number = 4 ;
number= number+ 2 ;
```

وما ينطبق على عملية الجمع ينطبق على جميع العمليات الحسابية الأخرى ....

مثال :

```
var number = 4 ;
number * = number;
```

وهي عملية ضرب وضعنا بداخل المتغير **(number)** قيمة **(4)** ثم اجرينا عملية الضرب على هذا المتغير الذي يحتوي على **(4)** وهي ضربه بنفسه وبالتالي الناتج هنا يساوي **(16).**

§ **جمل الشرط :**

قد نحتاج أثناء كتابتنا للبرنامج أن نضع جملة شرط أو أكثر به فبناتج جملة الشرط نستطيع أن نتحقق من المطابقة أو عدمها.. إذا هي مقارنة بين قيمتين قد يتطابقوا وعندها يكون الجواب **(True)** أي صحيح أو قد لا يتطابقوا وعندها يكون الجواب **(False)** أي خاطئ .

ولتوضيح الصورة، كأن نتحقق من كلمة السر هل هي صحيحة أم لا إذا هنا سوف يكون لدينا كلمتين سوف نقارن بينهما وهما كلمة السر المسجلة لدينا وكلمة السر التي سوف يدخلها المستخدم .

§ **لنتعرف إذا على جمل الشرط وأنواعها :**

١- الدالة الشرطية (if) :

في هذه الدالة نقوم بالتأكد من الشرط فإذا تحقق يتم الدخول الى داخل الدالة وبعد الإنتهاء من أوامرها ينتقل الى تنفيذ الاوامر التي تقع خارجها وإذا لم يتحقق سوف يتم الانتقال مباشرة الى تنفيذ الاوامر التى تقع خارج الدالة .

صيغتها:

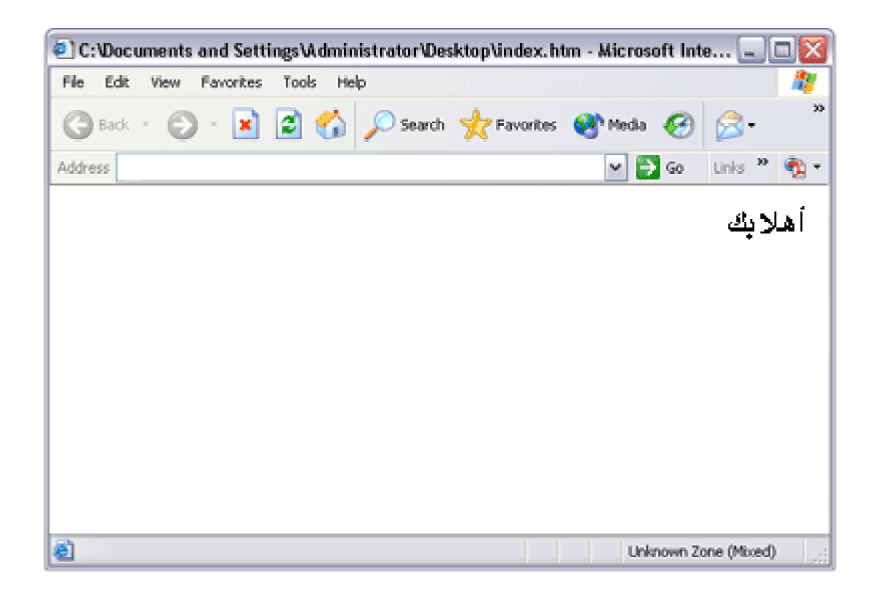
```
If ( الشرط )
  {
        إذا تحقق – نفذ الامر الذي بداخل الدالة
  }
        أوامر خارج الدالة
```

لنأخذ مثالا كاملا ونرى كيف :

```
<html dir="rtl">
    <head><title>الجافا سكربت</title>
      <script type ="text/javascript">
        var pass_2="okman";
        var  pass_user;
        pass_user = window.prompt("ادخل كلمة السر","كلمة السر");
          if (pass_2 == pass_user)
              {
                    document.write (" أهلا بك ") ;
              }
      </script>
    </head>
  <body></body>

</html>
```





في هذا المثال قمنا بتعريف متغير **(pass_2)** ووضعنا بداخله قيمة **(okman)** ثم قمنا بتعريف متغير **(pass_user)** ولم نضع بداخله قيمة على اساس ان المستخدم اي الزائر هو الذي سوف يضع القيمة وهنا سوف تخرج نافذة لتخبر المستخدم ان يدخل كلمة السر وعندها يضع المستخدم الكلمة وبالتالي سوف تخزن هذه القيمة بداخل المتغير **(pass_user)** وبعد ذلك ندخل بداخل جملة الشرط وهي الـ **(if)** وهنا سوف يقارن بين قيمة المتغير **(pass_2)** المخزنه مسبقا وبين قيمة المتغير **(pass_user)** الذي حدد قيمته المستخدم اذا جملة **(if (pass_2==pass_user))** تعني هل المتغير **(pass_2)** المخزن مسبقا يساوي قيمة المتغير **(pass_user)** المدخلة من قبل المستخدم .. هنا اذا كانت الاجابه نعم عندها سوف يطبع له جملة أهلا بك واذا كانت لا اي غير متساويتين اي ان كلمة السر الذي ادخلها المستخدم ليست متطابقة مع كلمة السر المخزنه مسبقا ... عندها لن يطبع شيء .. وفي هذا المثال افترضنا انه ادخل كلمة السر صحيحه" اقصد هنا في صورة المثال" ..

٢- الدالة الشرطية (if / else) :

في هذه الدالة الشرطية نقوم بالتأكد من الشرط وفي هذه الحالة سيكون لنا حالتين :

الاولى :

تحقق الشرط وبالتالي الدخول الى داخل الدالة لتنفيذ الاوامر التى بداخلها ثم الانتقال الى تنفيذ الاوامر التى تقع خارجها .

الثانية :

عدم تحقق الشرط وبالتالي الانتقال الى جملة **(else)** وتنفيذ مابداخلها ثم الانتقال الى تنفيذ الاوامر التى تقع خارجها .

صيغتها :

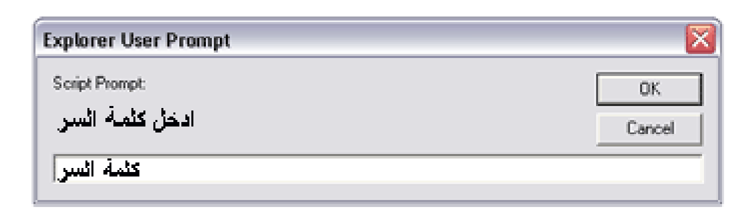
```
If ( الشرط )
    {
        إذا تحقق  نفذ مابداخلها
    }
else
    {
     نفذ هذا الامر إذا لم يتحقق الشرط
    }
          تنفيذ الاوامر الخارجية
```

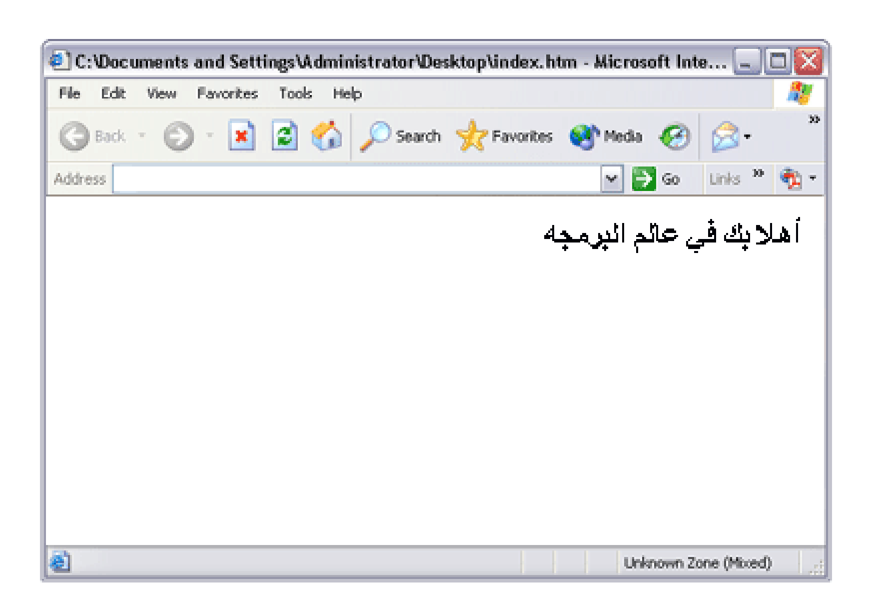
لنأخذ مثالا كاملا ونرى كيف ..؟

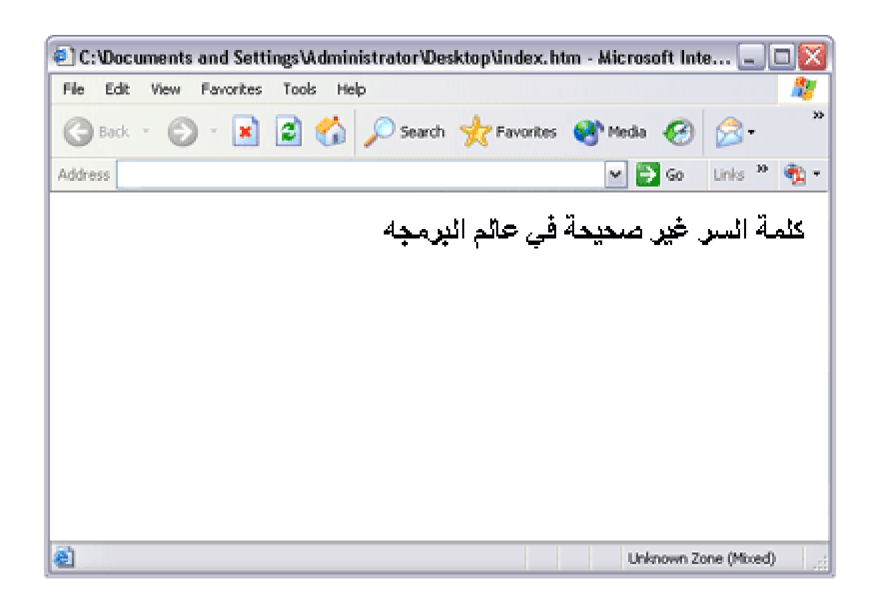
مثال :

```
<html dir="rtl">
  <head><title>الجافا سكربت</title>
    <script type ="text/javascript">
      var pass_2="okman";
      var  pass_user;
      pass_user = window.prompt("كلمةالسر"،"ادخل كلمة السر");
        if (pass_2 ==pass_user  )
              {
                  document.write ("  أهلا بك  " ) ;
              }
        else
              {
                  document.write ("  كلمة السر غير صحيحة   ") ;
              }
         document.write (" في عالم البرمجة  ") ;
    </script>
  </head>
 <body></body>

</html>
```







في هذا المثال قمنا بتعريف متغير **(pass_2)** ووضعنا بداخله قيمة **(okman)** ثم قمنا بتعريف متغير **(pass_user)** ولم نضع بداخله قيمة على اساس ان المستخدم اي الزائر هو الذي سوف يضع القيمة وهنا سوف تخرج نافذة لتخبر المستخدم ان يدخل كلمة السر وعندها يضع المستخدم الكمله وبالتالي سوف تخزن هذه القيمة بداخل المتغير **(pass_user)** وبعد ذالك ندخل بداخل جملة الشرط وهي الـ **(if)** وهنا سوف يقارن بين قيمة المتغير **(pass_2)** المخزنه مسبقا وبين قيمة المتغير **(pass_user)** الذي حدد قيمته المستخدم اذا جملة **(if (pass_2==pass_user))** تعني هل المتغير **(pass_2)** المخزن مسبقا تساوي قيمة المتغير **(pass_user)** المدخل من قبل المستخدم .. هنا اذا كانت الاجابه نعم عندها سوف يطبع له جملة أهلا بك واذا كانت لا اي غير متساوين اي ان كلمة السر الذي ادخلها المستخدم ليس مطابقة لكلمة السر المخزنه مسبقا ... عندها سوف يطبع الجمله التي في **(else)** وهي كلمة السر غير صحيحه.. وبعد ان يطابق ويرى اذا كانت صحيح ام لا ليحدد اي جملة يطبع سوف يخرج من دالة الـ **(if)** الى تنفيذ الجملة الخارجية اي التي خارج الدالة والتي ليس لها علاقة بماذا كانت المطابقة لا ام نعم وبتالي طباعة .. عالم البرمجة .. اذا هنا إما ان تكون الكلمة المدخله صحيحه وعندها سوف يطبع جملة اهلا بك ثم يطبع جملة عالم البرمجه او ان تكون الكلمة المدخله خطأ وعندها يطبع كلمة السر غير صحيحه ثم يطبع عالم البرمجه .. وفي هذا المثال افترضنا انه ادخل كلمة السر صح " اقصد هنا في صورة المثال "..

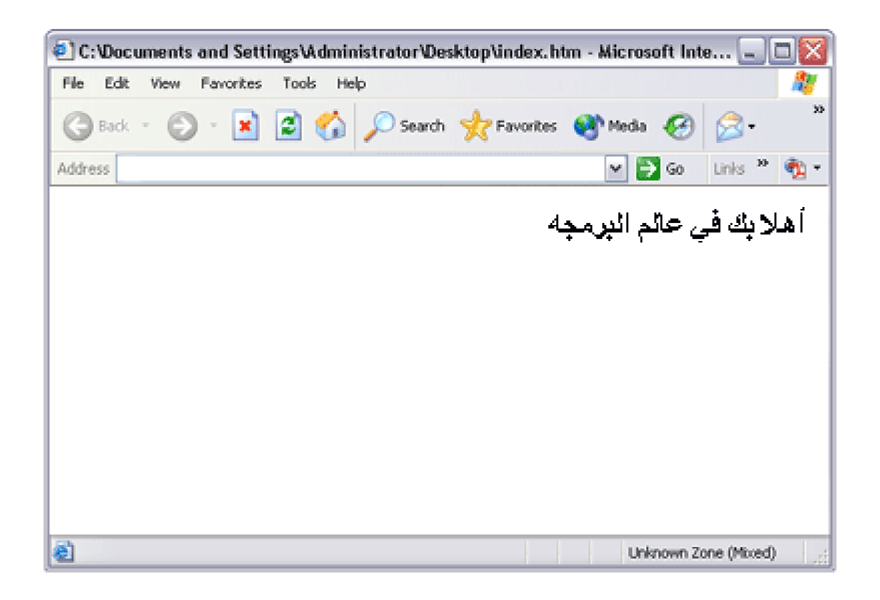
§ معلومات مهمة عن الدالة (if) :

١- قد يستخدم المبرمج شرط بداخل شرط بداخل شرط وتسمى هذه الحالة **(nested if)** فلا يزعجك ذلك فلكل منا أسلوبه الخاص في وصف الامور كما أنه قد يتحتم علينا أوقات أن نستخدم شرط بداخل شرط .

مثال :

```
<html dir="rtl">
   <head><title>الجافا سكريبت</title>
      <script type ="text/javascript">
         var name1 ="sami";
         var name2 ="mahmoud";
         if (name1 =="sami")
            {     if (name2 =="mahmoud")
                     document.write ("أهلا بك" ) ;
            }
         document.write (" في عالم البرمجة " ) ;
      </script>
    </head>
  <body></body>

</html>
```



في هذا المثال أضطررنا أن نستخدم شرط بداخل شرط وهذا الأسلوب يسمى (الأسلوب الهرمي اذا جاز التعبير) فهنا وضعنا شرطين بداخل بعض ... **لماذا ؟**
لنفرض أن لدينا عدة أشخاص جميعهم الإسم الأول سامي ولكن إسم الأب أي الإسم الثاني يختلف إذا هنا يجب أن نتحقق من الإسمين الأول والثاني إذا الخطوات التى نتبعها كما يلي :
نحضر جميع الأشخاص الذين إسمهم الأول (سامي) وهذا مافعلناه بالشرط الأول .

```
If ( name1 == " sami " );
```

فإذا كان الشرط قد تحقق أي أن أسم المستخدم (سامي) سوف يدخل الى الدالة ومن ثم سوف يجد شرطا آخر يسأله عن أسمه الثاني أي أسم الأب :

```
if ( name2 == " mahmoud ");
```

فإذا تحقق سوف ينفذ الامر الذي بداخل هذه الدالة . ومن ثم ينفذ الاوامر التي خارج الدالتين تماما .

لو لاحظت أنه في حالة عدم تحقق أي من الشرطين السابقتين سوف يخرج لتنفيذ ما بخارج الدالتين .

§ وقد يستخدم شخص أيضا دالة **(if)** داخل دالة **(if)** داخل دالة **(if)** وهكذا فلا أريد أن أدخلك في متاهات وخوارزميات معقدة . فكما قلنا لكلا منا أسلوبه فقد يكون حل مسألة بسطر فيأتي شخص ويحلها بأربعة أسطر بطريقة معقدة... إذا جميعنا يستطيع أن يصل الى الحل ولكن بأساليب مختلفة وقد يتطلب الامر ان نستخدم جمل **(if)** بداخل بعضها.

§ الدالة الشرطية (switch) :

عندما يكون لدينا عدة خيارات ونكون نريد أن نخرج بواحد منهم وهو الذي نريده أما ماهو الذي نريده من بين الخيارات والذي سوف نخرج به سوف يحدده المتغير الذي سوف ندخله والذي سوف يتفق مع واحدة من هذه الخيارات ويحققه .. أي هنا وكأننا نستخدم الجملة الشرطية (if) أو ماشابه ذلك اذا جاز العبير..
ويأتي أستخدام الدالة (switch) ليسهل العمل بشكل كبير وسوف نرى ذلك :

صيغتها :

```
switch (المتغير) {  بداية

      case :  الاحتمال الأول ;
            المطلوب لهذا الاحتمال

       case :  الاحتمال الثاني ;
             المطلوب لهذا الاحتمال

       case :  الاحتمال الثالث ;
            المطلوب لهذا الاحتمال
          .
          .
          .
       default :
                              المطلوب لهذا الاحتمال .. وهو يطبق في حالة عدم تطابق المتغير مع اي case ذكر ..
                 } " نهاية " اي اغلاق
```

لنأخذ مثال ونحلله.. والذي من خلاله سوف نفهم ماقصدناه في عباراتنا السابقه.. فأنا أعرف جيدا عزيزي المتدرب أنه كلام مبهم وغير واضح لك الى الان.

مثال :

```
<html dir="rtl">
  <head><title>الجافا سكربت</title>
   <script type = "text/javascript">
    var country ;

   country=window.prompt( " أدخل الدولة لتعرف عاصمتها ", "0");

   switch ( country ) {

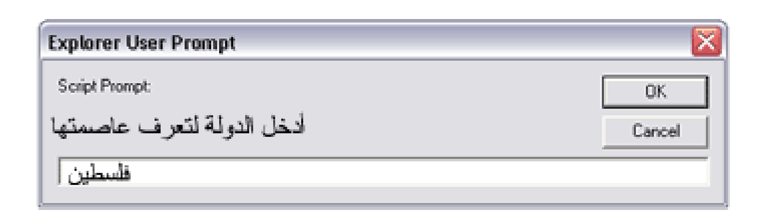
     case " فلسطين " :
          document.writeln("<h3> عاصمتها القدس الشريف</h3>") ;
               breack ;

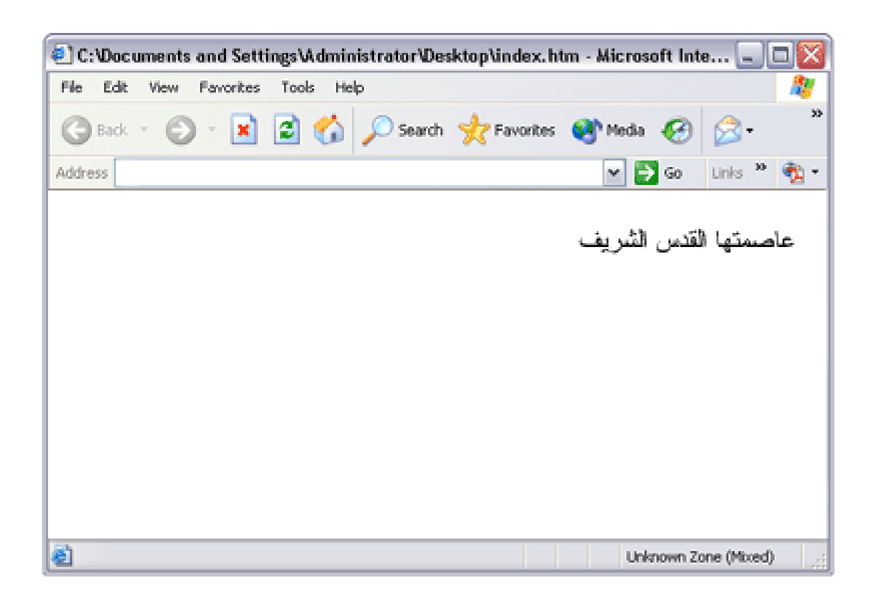
     case " العراق "  :
          document.writeln("<h3>  عاصمتها بغداد </h3>") ;
              breack;

     case " السودان " :
         document.writeln("<h3> عاصمتها الخرطوم </h3>") ;
         breack ;

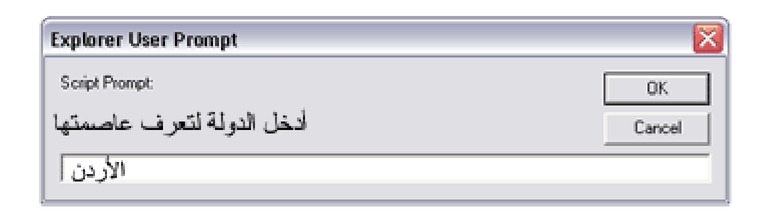
     case " الوطن العربي " :
         document.writeln("<h3>  من المحيط الى الخليج </h3>") ;
          breack ;

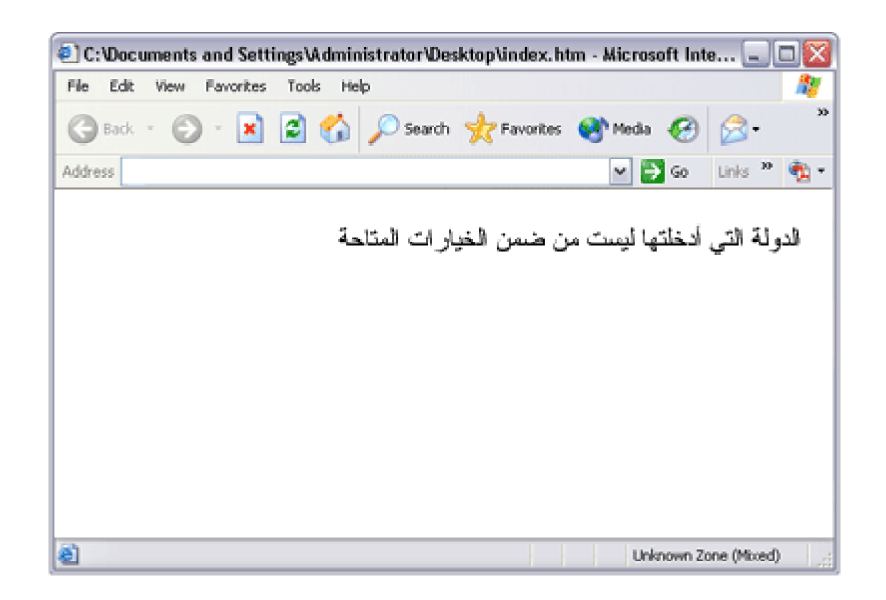
    default :
          document.writeln("<h3>     الدولة التي أدخلتها ليست من ضمن الخيارات المتاحة </h3>") ;
                          }
      </script>
    </head>
  <body></body>
</html>
```





§ في حالة ادخال داله ليست ضمن الدول المذكوره في ال(**case**) :





لنحلل المثال سويا فبه نعرف مابهم علينا :

في البداية عرفنا المتغير **(country)** وأسندناه الى جملة **(window.prompet)** لكي يدخله المستخدم بنفسه من لوحة المفاتيح أي انه سوف يدخل أسم دولة معينة ... وهنا سوف نأتي الى وضع المتغير في دالة **(switch)** كالتالي :

```
switch ( country )
```

ثم نضع الخيارات التي نريدها وسوف يقارن المتغير الذي أدخلنا قيمته بما هو موجود بجانب كلمة **(case)** كالشكل التالي :

```
case " السودان " :
```

وفي حالة مطابقته لأي واحدة من الخيارات التي وضعناها سوف ينفذ الجمل التي جاءت ضمن ال **(case)** الذي توافق معه وبالتالي طباعة الجملة والخروج من دالة **(switch)** ولهذا نضع بنهاية كل **(case)** أمر **(break)** ليتوقف , لاننا هنا نريد أن نخرج بقيمة واحدة وهي احدى الخيارات التي وضعناها " جرب أن لا تضع **(break)** وأنظر ماذا سيحدث فبالتجربة يحدد المفهوم " .

ولاحظ أننا وضعنا في النهاية الدالة وقبل اغلاقها **(default)** وهنا نعني أنه في حالة أن المستخدم أدخل قيمة للمتغير وكانت ليست ضمن الـ **(case)** سوف يخرج المستخدم بالجملة والامر الذي سوف نضعه بعد **(default)** ثم ننهي الدالة بوضع النهاية **({)** فما رأيك عزيزي المتدرب أن نأخذ المثال وكأننا مستخدمون وننفذه الان على غير ما تم ادخاله في الصور السابقة من دولة فلسطين .

نفرض خرجت نافذة **(prompt)** وأدخلنا الدولة العراق مثلا فهنا سوف يأتي البرنامج ويأخذ كلمة العراق والتي هي اسندت الى المتغير ويضعها في دالة **(switch)**

```
switch ( country )
```

فيدخل بداخل دالة **(switch)** فيأتي الى الـ **(case)** الاول والتي هي :

```
case " فلسطين " ;
```

فيجد أن كلمة العراق لاتتطابق كلمة فلسطين إذا ليس هذا هو الخيار . فينتقل الى الـ **(case)** الآخر ...

فيجد أن المتغير به كلمة العراق يطابق هذا الـ **(case)** أي **(العراق = العراق)** فينفذ الجملة التي بداخل هذا الـ **(case)** التي هي :

```
document.writeLn( "<h3>  عاصمتها بغداد </h3> ") ;
```

وبالتالي طباعة الجملة التالية : " عاصمتها بغداد "

وبعد جملة الطباعة هذه يأتي أمر **(break)** أي توقف وبالتالي الخروج من دالة **(switch)** كاملتا دون المرور بالـ **(case)** التي تأتي بعدها .

- وأيضا لو فرضنا أن شخص أدخل أي دولة غير موجودة ضمن الخيارات لتكن الاردن أو السعودية أو الكويت أو الامارات ... الخ سوف يطبع هنا الجملة التي ضمن الامر **(default)** أي :

```
document.writeln( "<h3>  الدولة التي أدخلتها ليست من ضمن الخيارات المتاحة </h3> ") ;
```

§ الذي أريد ان أوصله لك في النهاية هو شئ واحد وهي أننا ندخل المتغير في جملة **(switch)** للمقارنة مع الخيارات الموجودة بداخلها وعند مطابقة المتغير مع احدى الخيارات أي النتيجة **(true)** سوف ينفذ الجملة المتعلقة بهذه المطابقة مع العلم أن واحدة فقط من هذه الخيارات تعطي **(true)** والباقية **(false)** وعندما يكون كل الخيارات ليست مطابقه سوف ينفذ ما بداخل **(default)** اي ان جميع ال **(case)** غير مطابقة **(false)**.. اذا ياتي عمل ال **(default).**

§ حلقات التكرار :
ونعني بها تكرار جملة معينة عدة مرات فحلقة التكرار سوف تستمر مادام الشرط متحققا ويتم التحقق من الشرط في بداية الحلقة او نهايتها .

§ قد تتساءل مافائدة أستخدام حلقات التكرار ؟
نجيب بذلك . . .
أفترض انك تريد طباعة جملة معينة ١٠ مرات فأنك سوف تحتاج الى حلقة التكرار فلو لم تستخدمها لإضطررت لكتابة كود الطباعة ١٠ مرات ولكن بحلقة التكرار لايأخذ منك سوى كتابة سطرين من الكود .
وكذلك الحال لو أردت طباعة أسماء جميع من هم مسجلين لديك بقاعدة بيانات الموقع .

§ أنواع حلقات التكرار :
١- حلقة التكرار (for) :
وفي هذه الحلقة يكون الشرط هو الذي يحدد طول الحلقة فما أن يصل الى نهايتها حتى يتوقف أي فشل الشرط ولكن مادام الشرط متحقق سوف يدخل الى داخل الحلقة وينفذ مابداخلها .

صيغتها :

```
for ( بداية الحلقة ; شرط الحلقة ; مقدار الزيادة )

   {
              نفذ الامر
   }
```

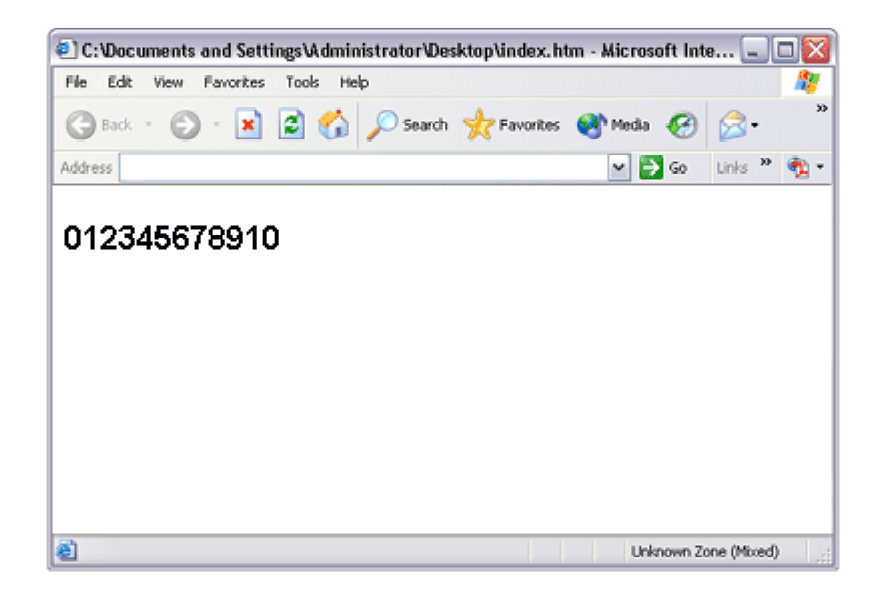
شرط الحلقة : هو الذي سوف يحدد طول الحلقة (عددالتكرار)

مثال :

```
<html >
  <head><title>الجافا سكربت</title>
    <script type = "text/javascript">

      for (var i = 0 ; i<= 10 ; i ++ )
           {
           document.write ( i ) ;
            }
   </script>
  </head>
 <body></body>
</html>
```

ملاحظة : انا عرفت المتغير i في داخل حلقة التكرار تستطيع تعريفه في الخارج كما في السابق ووضعه مباشرة ...



لنحلل هذا المثال سويا :

هنا أول شي عرفنا المتغير وهو **(i)** وجعلنا قيمته الأبتدائية **(i=0)** أي أنه سوف يبدأ من **(0)** .

ثم حددنا الشرط والذي هو **(i < = 10)** أي أننا سوف ندخل الى داخل الحلقة وننفذ مابداخلها من أوامر مادامت قيمة **(i)** أقل أو تساوي **(10)** أي أنها لو وصلت الى **(11)** سوف يخرج من الحلقة ولن يدخلها إذا هنا حددنا نهاية **(i)** وهي **(10)** . أي أن التكرار سوف يكون ١٠ مرات ولاننا بدأنا من الصفر سوف يكون ١١ .

ثم حددنا مقدار زيادة المتغير **(i)** وحددناه بأن يزيد بـ **(1)** بكل مرة يدخل فيها الى الحلقة .

*** نأتي للتطبيق العملي للمثال " أي مايحدث بداخل البرنامج " :**

لنكتب الصيغة ونتتبعها :

```
for ( var i = 0 ; i < = 10 ; i ++ )
```

نبدأ بقيمة **(i)** كما ذكرنا وهي **(0) .**

ننتقل الى الخطوة التالية التى تليها وهي الشرط **(i < = 10)** وهنا يسأل هل **(i)** أقل أو تساوي **(10)** سوف تكون الاجابة نعم .

ثم ينتقل الى للخطوة التى تليها وهي **(i++)** أي يعني ذلك بعد الدخول الى حلقة التكرار والرجوع الى**(for)** زيد **(i)** بمقدار **(1)** .

دخلنا الى الحلقة الان سوف يطبع قيمة الـ **(i)** والتى هي **(0)** ثم يعود الى**(for)** ولكن بعد زيادة قيمة **(i)** بقيمة **(1)** لتصبح قيمتها **(1).**

الان سوف يكون مافعلناه بالخطوات السابقة ولكن هذه المرة سوف تتغير قيمة الـ **(i)** الأبتدائية الى **(1)** بدلا من **(0)** .

- للتوضيح أكثر نعيد الخطوات :

- الان سوف يسأل هل **(1)** أقل أو يساوي **(10)** سوف تكون الأجابة نعم فالـ **(1)** أصغر من **(10)** .
- ثم ينتقل الى الخطوة التى تليها وهي **(i++)** أي يعني ذلك بعد الدخول الى حلقة التكرار وتنفذ مابداخلها والرجوع الى الـ **(for)** زيد قيمة **(i)** بمقدار **(1)** أي تصبح **(2)** .
- وهنا سوف يعود الى الـ **(for)** وسوف تصبح قيمة ال **(i)** الأبتدائية "اذا جاز التعبير" وهي **(2)** ويسأل الشرط والخطوات التى ذكرناها سابقا ويستمر حتى يصل ال قيمة **(11)** وهنا سوف يحدث التالي :

§ سوف يجعل قيمة الـ (i) الأبتدائية (11) ثم ينتقل الى الخطوة التى تليها بالـ (for) وهي الشرط (10 = > i) سوف يسأل هل قيمة (i) والتى هي الان (11) أقل أو تساوي (10) سوف تكون الاجابة لا إذا سوف يخرج من الحلقة ولن ينفذ الجملة التى بداخلها .

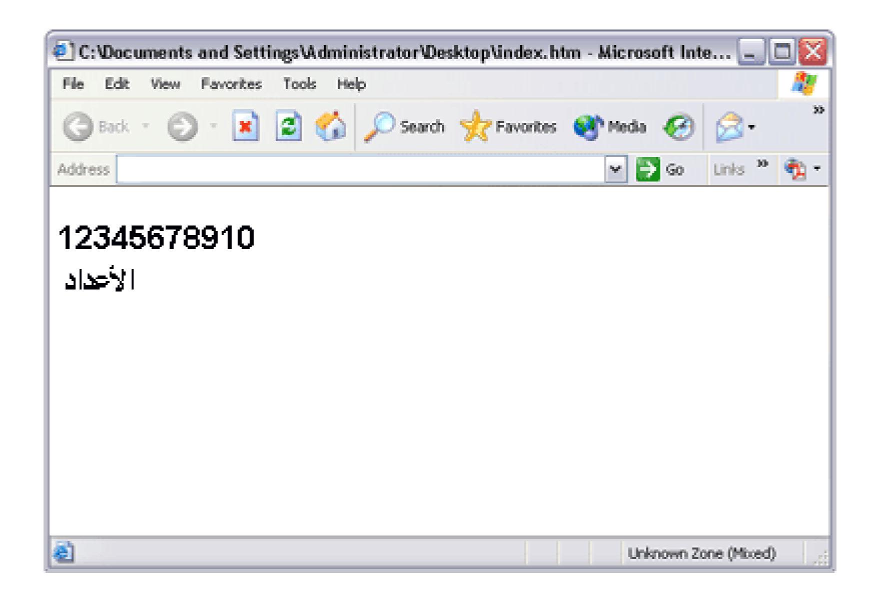
§ يعني بإختصار . . .

سوف يستمر بتنفيذ جملة الطباعة التى بداخل حلقة الـ (for) مادام الشرط متحققا ولن يخرج منها إلا بعد فشل الشرط . فلو كان يوجد جملة تلي حلقة التكرار كالمثال الذي سوف أطرحه الان لن تنفذ الا بعد الخروج من حلقة التكرار .

مثال :

```
<html >
   <head><title>الجافا سكريبت</title>
     <script type = "text/javascript">
         for (var i = 1 ; i <=10; i ++ )
             {
                 document.write ( i ) ;
             }
         document.write (" <br />  الاعداد");
    </script>
   </head>
  <body></body>

</html>
```



ملاحظة : سوف احاول شرحه بطريقه اكثر تقريبا ...
هنا في هذا المثال سوف يبدأ قيمة الـ (i) وهي (1) وسوف يكون الشرط أن مادام قيمة الـ (i) أقل من أو يساوي (10) أطبع وعند فشل الشرط أخرج من حلقة التكرار وإذهب لتنفيذ الجملة التى تلي حلقة التكرار وهي طباعة جملة (الأعداد)

هنا يكون التنفيذ على النحو التالي : رسمة لتوضيح

```
i =1 à  i ++ à i = 2
i =2 à  i ++ à i = 3
i =3 à  i ++ à i = 4
i =4 à  i ++ à i = 5
i =5 à  i ++ à i = 6
i =6 à  i ++ à i = 7
i =7 à  i ++ à i = 8
i =8 à  i ++ à i = 9
i =9 à  i ++ à i =10
```

```
i = ' 10 ' à  i ++ à i =11
```

فنحن محددين أنه آخر قيمة تطبع هي **(10)** فبعد أن يزيد قيمة **(i)** سوف يفشل الشرط وبالتالي سوف يخرج من حلقة التكرار لان قيمة **(i)** اصبحت **(11)** اذا هي ليست اقل من **(10)** ثم ينتقل الى الجملة التى تلي حلقة التكرار والتى هي خارج حلقة التكرار وهي طباعة " الأعداد "

```
document.write ( " <br /> الاعداد);
```

إذا النتيجة هي طباعة مايلي :

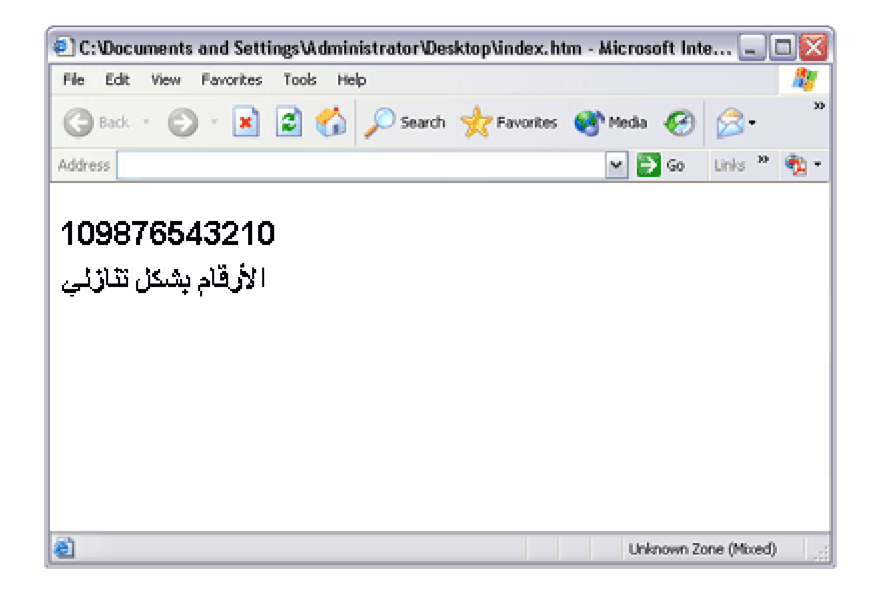
12345678910
الاعداد

لأطرح لك مثالا وأترك تحليله لك ...
لنفرض أننا نريد طباعة الارقام ولكن بالعكس أي من **(10)** الى **(0)**

تكون صيغته كالتالي :

```
<html >
    <head><title>الجافا سكريبت</title>
      <script type = "text/javascript">
        for (var i =10 ;  i>=0 ; i--)
          {
              document.write( i ) ;
          }
        document.write ( " <br /> الأرقام بشكل تنازلي")
     </script>
   </head>
  <body></body>

</html>
```



النتيجة :

10987654321
الأرقام بشكل تنازلي

وهذا المثال يسمى طباعة **' reverse '** للقيم اي طباعة عكسية.

§ **حلقة التكرار (While) :**
وفي هذه الطريقة سوف يستمر يتنفيذ مابداخل الحلقة مادام الشرط متحقق في كل مره تريد بها الدخول سوف يتحقق من الشرط أولا فإذا تحقق تقوم بالدخول الى داخل الحلقة وتنفيذ مابداخلها الى أن يفشل ونخرج من الحلقة مع العلم انه اذا فشل من البدايه اذا لن يدخل اطلاقا الى داخل حلقة التكرار ولكن اذا يوجد اوامر خارج حلقة التكرار اي بعد حلقة **while** سوف ينفذها .

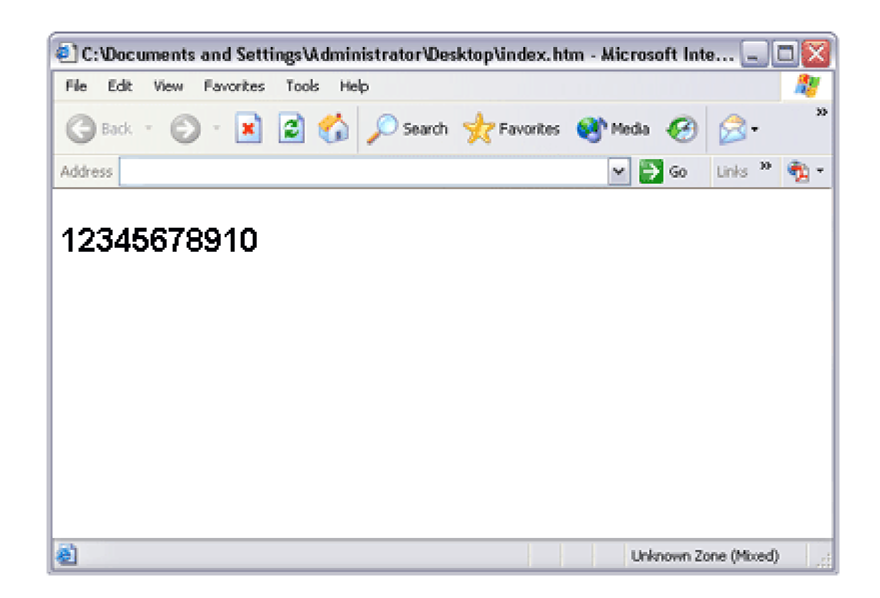
صيغتها :

```
While ( نفذ مابداخل الحلقة مادام الشرط متحققة )
{
      نفذ الامر
  i ++    مقدار الزيادة
}
```

قبل الخوض بمثال يجب أن تحتوي جملة الـ (while) على مايلي :
١- متغير نضعه بالشرط لكي نتحقق من صحة الشرط .
٢- يجب وضع قيمة إبتدائية لهذا المتغير قبل جملة الـ **(while)** .
٣- يجب أن نذكر هذا المتغير ومقدار زيادته بداخل حلقة الشرط سواءا قبل تنفيذ الجملة التي بداخل حلقة التكرار أو بعدها .

مثال :

```
<html >
 <head><title>الجافا سكريبت</title>
  <script type = "text/javascript">
   var   i = 1 ;
   while (i <= 10 )
     {
      document.write ( i ) ;
     i ++ ;
     }
  </script>
 </head>
 <body></body>
</html>
```



لنحلل المثال سويا :
أول شئ عرفنا **(i)** كمتغير ووضعنا به القيمة الأبتدائية **(1)** قبل أن نضعه في الشرط **(i <= 10)** .
وبعد ذلك ذهبنا الى جملة **(while)** فسألنا الشرط هل قيمة **(i)** أقل أو تساوي **(10)** فكانت الاجابة نعم فالـ **(1)** أقل من **(10)** وبما أنه تحقق
الشرط فسوف ندخل بداخل الحلقة ...

```
While ( 1 <= 10 )
```

فدخلنا بداخلها ونفذنا الجملة التى بداخل حلقة التكرار التى هي طباعة قيمة **(i)** والتى هي **(1)** .

```
document.write ( i ) ;
```

ثم أنتقلنا الى الجملة التى تليها والتى هي **(++i)** وتعني زيد قيمة **(i)** بمقدار **(1)** فأصبحت قيمة **(i)** تساوي **(2)**.

تم عاد مرة أخرى الى **(while)** فسألة الشرط هل قيمة **(i)** أقل أو تساوي **(10)** فكانت الاجابة بنعم فنحن عدنا للـ **(while)** بقيمة **(i)** بعد زيادتها بـ **(1)** أي **(2)** .

```
While ( 2 <= 10 )
```

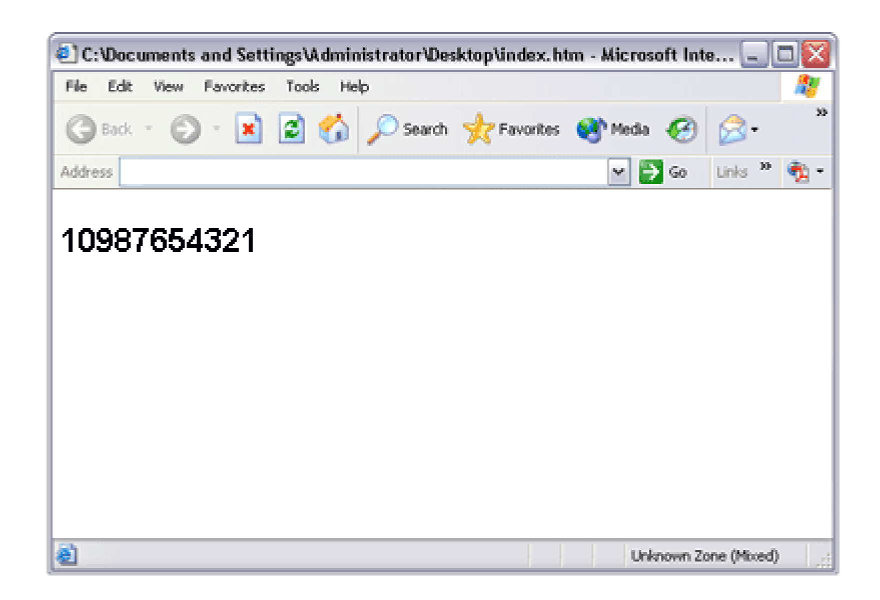
فبعد تحقق الشرط دخلنا الى داخل الحلقة فوجدنا أمر الطباعة قيمة **(i)** وهو **(2)** .
ثم أنتقلنا الى الجملة التى تليها والتى هي **(i++)** " أي زيد قيمة **(i)** بمقدار **(1)** وقيمة **(i)** عندنا الان **(2)** زيدها بواحد لتصبح **(3)** " .

ثم نعود الى الـ **(while)** مرة أخرى ومعنا قيمة الـ **(i)** الجديدة وهي **(3)** ويتحقق من الشرط مرة أخرى وهكذا نستمر **10** مرات الى أن تصل قيمة **(i)** الى **(11)** وهنا سوف يحدث التالي :

يأتي الجملة الـ **(while)** فسألة الشرط هل قيمة **(i)** التى هي **(11)** أقل أو تساوي **(10)** فيكون الجواب لا فيخرج من حلقة التكرار ولايدخل بداخلها . وإذا كان هناك جملة تلي حلقة التكرار سوف ينفذها بعد خروجه من حلقة التكرار .
(لاحظ جملة تلي حلقة التكرار وليس بداخله حلقة التكرار) .
مثال اخر (لنجعلك تقارن بين **(while)** و **(for)**) . .

مثال (1) :

```
<html >
     <head><title>الجافا سكربت</title>
   <script type = "text/javascript">
        var  i = 10 ;
       while (i>= 1 )
           {
              document.write ( i ) ;
                i -- ;
            }
   </script>
  </head>
 <body></body>
</html>
```



§ حلقة التكرار (Do while) :

وفي هذه الطريقة سوف يستمر بتنفيذ مابداخل الحلقة مادام الشرط متحقق و هنا سوف يدخل الى داخل الحلقة ومن ثم ينفذ الامر الذي بداخلها وبعد تنفيذها ينتقل ليتحقق من الشرط فإذا تحقق يعود مرة آخرى الى الحلقة وينفذ الامر مرة أخرى وإذا لم يتحقق يخرج من حلقة التكرار ولن يعود لها .

صيغتها :

```
do {
        نفذ الامر
        مقدار الزيادة
   }
while (  عد الى داخل الحلقة مرة أخرى ونفذ مادام الشرط يتحقق )
```

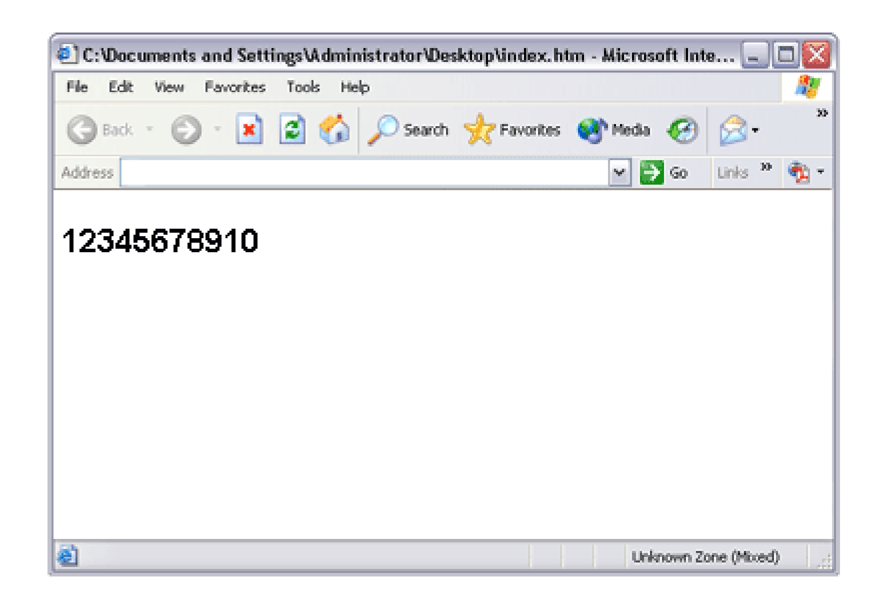
قبل الخوض في مثال يجب أن تحتوي جملة (do while) على مايلي :

١- متغير نضعه بالشرط لكي نتحقق من صحة الشرط .
٢- يجب وضع قيمة إبتدائية لهذا المتغير قبل جملة الـ **(do while)**
٣- يجب أن نذكر هذا المتغير ومقدار زيادته بداخل حلقة الشرط سواءا قبل تنفيذ الجملة التى بداخل حلقة التكرار أو بعدها .

مثال :

```
<html >
  <head><title>الجافا سكربت</title>
    <script type = "text/javascript">
       var  i = 1 ;
       do {
              document.write( i ) ;
              i ++ ;
          }
       while ( i <= 10  )
    </script>
  </head>
 <body></body>

</html>
```



لنحلل هذا المثال سويا :
وضعنا قيمة إبتدائية للـ (i) وهي (1) ثم دخلنا الى داخل (do while) ثم توجهنا الى الجملة التى تليها هي جملة الطباعة " أي طباعة قيمة (i) " إذا سوف يطبع (١) ثم وجدنا الامر (i++) اي زيادة قيمة (i) أي أصبحت قيمة (i) تساوي (2). ثم انتقلنا الى جملة الشرط التي حدد هل سوف نعود الى (do) أم لا . فوجدنا الشرط يسألنا هل قيمة (i) أي (2) أقل أو تساوي (10) فكان الجواب نعم إذا عدنا للـ (do) مرة أخرى ثم نفذنا مابداخلها فزدنا قيمة (i) بواحد فأصبحت (3) ثم انتقلنا الى الشرط وقد نحقق وكانت الاجابة بنعم أي أن الـ (3) أقل من (10) وهكذا استمرينا الى أن نصل لقيمة الـ (i) يساوي (11) وبالتالي عدم العودة الى (do) والخروج من حلقة التكرار .

قد تتساءل مالفرق بين الـ (while) والـ (do while) ؟
نحدد ذلك الفرق . . .

كما في (while) نتحقق من الشرط قبل الدخول الى الحلقة أي أننا لاننفذ أي شئ بداخلها مادام الشرط لم يتحقق وهذا أمر طبيعي لاننا لم ندخل الى الحلقة أصلا فكيف نعرف مابداخلها وننفذه .
أما بالـ (do while) كنا ندخل الى الحلقة وننفذ أمر ثم نفحص الشرط ولكن بعد أن نكون قد نفذنا هذا الامر ويجب التنبيه هنا " في حالة عدم تحقق الشرط لن نعود مرة أخرى الى (do) .
إذا الفرق هو أن بالـ (do while) ينفذ على الاقل أمر واحد في داخل حلقة التكرار حتى لو كان الشرط غير متحقق على العكس الـ (while) الذي لاينفذ أي امر مادام الشرط غير متحقق .

مثال توضيحي (لتوضح الفكرة بشكل عملي) :

```
var  i =5 ;
do
   {
      document.write ( i ) ;
      i++ ;
   }   while ( i < 3 )
```

```
var  i =5 ;
while ( i < 3 )
  {
     document.write ( i ) ;
      i ++ ;
   }
```

في مثال الـ (while) البرنامج لن يطبع شيئا وذلك لانه سألته جملة الشرط قبل الدخول هل الـ (5) أقل من (3) فكان الجواب لا . وهنا لن يدخل الى الحلقة وبالتالي عدم تنفيذ مابداخلها أي عدم طباعة قيمة (i)

في مثال الـ (do while) سوف يطبع البرنامج فقط مرة واحدة قيمة (i) ثم يتوقف فبعد طباعة قيمة (i) سوف يسأل عن الشرط وهو هل (6) أصغر من (3) فسوف تكون الاجابة لا وبالتالي عدم العودة الى جملة (do) أي عدم الدخول الى حلقة التكرار والاستمرار بالطباعة أي الخروج .

وفي هذا المثال نكون قد وضعنا الفرق بينهما ...

- **اذا نتيجة كود ال (while) أنه لن يطبع شيء ....**
- **ونتيجة ال do while هو طباعة فقط رقم (5) ....**

قد تتساءل عزيزي المتدرب مالفائدة من أخذنا الـ (while) والـ (do while) مادام أنهما يعملوا عمل الـ (for) ؟

عزيزي المتدرب نحن نذكر لك كل صيغ التكرار في لغة الجافا سكربت " والمستخدمة في جميع لغات البرمجه " ولك حرية الاختيار بإستعمال أي واحدة منها فقد تجد أن (for _ loop) هي أفضل لك ولكن عند غيرك تجد الـ (while) أو (do while) هي أفضل له وقد يتحتم عليك إستخدام أي واحدة منها مثل (do while) في كتابت برنامجك فهذا راجع لك وما مطلبك من برنامجك فكما قلنا لكلا منا أسلوبه وطريقته الخاصة بكتابة البرنامج .

§ الامر (break) و (continue) :

نبدأ بالامر (break) :
قد نستخدمها في الـ (while) أو الـ (do – while) وقد نستخدمها بالـ (switch) .
أما لماذا نستخدمها ؟

لنفرض أن لديك بيانات مخزنة في قواعد البيانات ولكن تريد فقط جزء من هذه البيانات أن تعرض وليس جميعا فهنا أستخدم (break) لكي أخرج مباشرة من داخل حلقة التكرار حتى لو لم أنتهي من الحلقة .

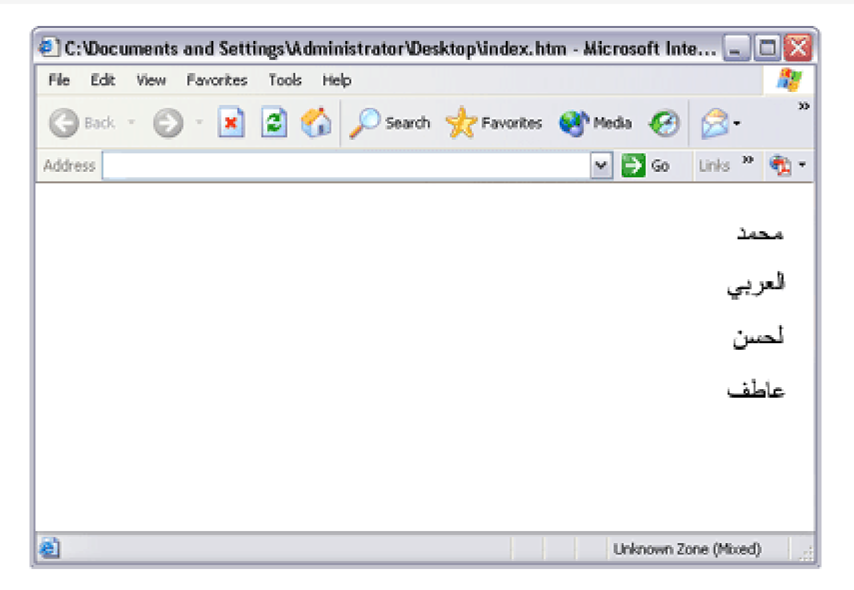
صيغتها :

```
break ;
```

لنأخذ مثالا :

```
<html dir="rtl">
      <head><title>الجافا سكريبت</title>
          <script type = "text/javascript">
               var x = new Array("محمد","العربي ","لحسن ","عاطف","محمود","ثامر ","خالد") ;
               for(var i=0 ; i <x.length ; i++)
                   {
                        if(i == 4)
                        break ;
                        document.writeln(x[i] + "<br />") ;
                    }
           </script>
       </head>
    <body></body>

</html>
```



لنحلل المثال سويا ..

عرفنا المتغير (x) والذي هو عبارة عن مصفوفة تحتوي بداخلها على (7) أسماء . ثم انشأنا حلقة التكرار (for) وبدأناها من (0) الى (i < x.lengh) أي أقل من العدد (7) .
وعزيزي المستخدم الامر الطبيعي والبديهي أن يطبع جميع الأسماء التي بالمصفوفة لكن هنا الذي سوف يحدث أنه سوف يطبع فقط أربعة أسماء . ويخرج خارج حلقة التكرار لماذا ...

لأننا وضعنا شرط داخل حلقة التكرار وهو **(if i == 4)** وبعده وضعنا **(break)** أي أننا حددنا إذا كانت قيمة **(i)** مساوية للعدد **(4)** يتوقف من حلقة التكرار وأيضا لاينفذ أمر **(document.writeln)** أي يخرج عند وصوله نقطة **(break)** ويخرج حتى لو لم تنتهي حلقة التكرار .
إذا هنا فقط سوف يطبع الأسماء الأربعة الاوائل والتي هم (**محمد _ العربي _ لحسن _ عاطف**) ولا تنسى اننا بدأنا من الصفر .

الامر (continue) :
هذا الامر عمله هو القفز عن قيمة معينة نحن نحددها أي يعمل **(skip)** ... لنفرض أن لدينا في قاعدة البيانات مثلا أو في مصفوفة سبعة أسماء ولكن الأسم رقم **(4)** لانريد طباعته فقط هو ... أما البقية نريد طباعتها إذا هنا الذي سوف يحل لنا هذا الامر هو **(continue)** .

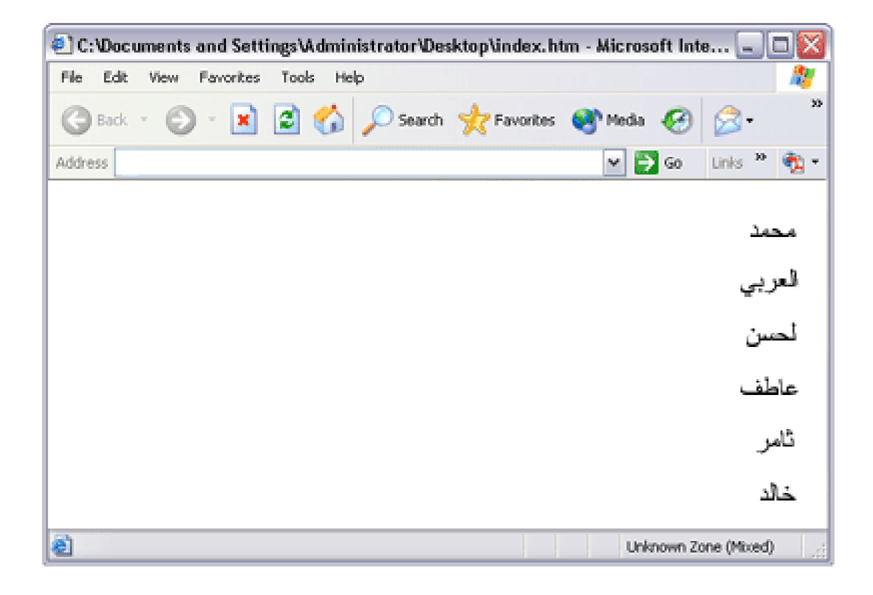
صيغتها :

```
continue ;
```

لنأخذ مثالا :

```
<html dir="rtl">
    <head><title>الجافا سكريبت</title>
        <script type = "text/javascript">
            var x = new Array("محمد","العربي","لحسن","عاطف","محمود","ثامر","خالد") ;
            for(var i=0 ; i <x.length ; i++)
              {
                 if(i == 4)
                 continue;
                document.writeln(x[i] + "<br />") ;
              }
        </script>
    </head>
  <body></body>

</html>
```



لنحلل المثال سويا :

عرفنا المتغير **(x)** والذي هو عبارة عن مصفوفة تحتوي بداخلها على **(7)** أسماء . ثم انشأنا حلقة التكرار **(for)** وبدأناها من العدد **(0)** الى **(i < x.lengh)** أي أقل من العدد **(7)** .
وعزيزي المستخدم الامر الطبيعي والبديهي أن يطبع جميع الأسماء التي بالمصفوفة لكن هنا الذي سوف يحدث أنه سوف يطبع جميع الأسماء ماعدا أسم (محمود) . **لماذا ؟**
نحن هنا كتبنا شرط والذي وهو **(if i == 4)** وبعده كتبنا **(continue)** أي كأننا نقول هنا عندما تصل قيمة **(i)** للعدد **(4)** نفذ **(continue)** أي الموقع الرابع من المصفوفة ..تجاهله ولا تطبعه وانتقل الى الذي يليه اي ارجع وتجاوز عن امر الطباعة... فهنا سوف يطبع الأسماء الاربعة الاولى والتي هي (محمد _ العربي _ لحسن _ عاطف) أي أننا وصلنا الى الموقع الرابع من المصفوفة ثم تصبح قيمة **(i)** هي **(4)** الأسم (محمود) فينفذ الامر **(continue)** ويعمل **(skip)** أي تجاهل وقفز عن هذا الأسم ثم يستمر في طباعة بقية الأسماء .

ملاحظة :
قد يتطلب منك برنامج معين حسب وظيفته الى وضع **for** او **while** بداخل **for** او **while** اي **nested loop** كما في ال **if** تذكر

§ الدوال (function) :
هنا سوف نتحدث عن كيفية إنشاء دالة أو أكثر خاصة بك بداخل برنامجك ولكن قد تتساءل ماهي الدوال سوف نجيبك بالتالي . .

عزيزي المتدرب الدالة هي من الاوامر والمتغيرات تكون يشكل مستقل بحيث نضعها داخل البرنامج ولا ينفذها البرنامج الا عندما يتم أستدعائها وهذا مايصطلح عليه بـ **(call function)** أما كيفية أستدعائها فيتم عن طريق كتابة أسم الدالة وأرسال المتغيرات لها " إن وجدت " .

بعد ماذكرناه قد تتساءل عزيزي المتدرب لماذا أستخدم هذه الدوال وماقيمتها بالنسبة لي أي ماهي أهميتها نجيبك بما يلي . . .

عزيزي المتدرب تستطيع أن لاتستخدم الدوال وتكتب برنامجك كالمعتاد ومن غير دوال ولكن تعرف ماذا سيحدث . . .

سوف تحدث سلبيات عدة منها :
أضاعة كثيرا من وقتك أضافة الى أن البرنامج سوف يحتوي على كود طويل للغاية فما بالك سوف تكون سرعة تنفيذ هذا الكود كذلك سوف يكون برنامجك صعب التطوير أعرف أنك قد أختلط عليك الامر لنطرح مثالا لتوضيح ما بهم لديك . . .

لنفرض أنك تريد أن تطلب من المستخدم أن يدخل رقمين والبرنامج الذي صممته يخرج ناتج جمعهما وتريد أن يكرر هذه العملية عشرات المرات فهنا تستطيع أستخدام الدوال فما عليك سوى أرسال الرقمين الذين يدخلهما في كل مرة الى الدالة. ولشدك للموضوع أكثر لا أريد أن أتكلم فقط على مثال الأرقام لأعطيك مثالا نستخدمه في المواقع ألست عندما تكون مشترك في موقع ما تدخل أسم المستخدم وكلمة السر للدخول الى الموقع وكذلك الحال لكثير من المشتركين في الموقع . إذا العملية تتكرر (أي فحص أسم المستخدم وكلمة المرور) عشرات المرات إذا لماذا لاأعمل دالة خاصة يدخل فيها المستخدم كلمة المرور وأسم المستخدم فتفحصها ونتأكد أنه مسجل أم لا.
عزيزي المستخدم قد لا يتوضح الامر لديك الى الان ولكن لاعليك فالعيب ليس منك فسوف تتوضح لك الامور بعد قليل إنشاء الله . . .

الصيغة العامة التى تكتب بهاالدوال (function) :
أي المتغيرات التى سوف تحدث عليها العمليات داخل الدالة

```
function   أسم الدالة وتختار أي أسم تريد غير محجوز   ( المتغيرات التى سوف ترسل للدالة )

   {   بداية الدالة

      العمليات التى تتم بداخل الدالة

      return   النتيجة المعادة ;

   }   نهاية الدالة
```

§ **ملاحظة :**
يمكن أن تأتي صيغة كتابة الدالة بشكل آخر وهو كالتالي :

```
Function أسم الدالة وتختار أي أسم تريد ( المتغيرات التى سوف ترسل للدالة )
  {
      return   العمليات التى سوف تتم ومن ثم إرجاع النتيجة ;
  }
```

كما ترى لم نغير أي شئ فقط الذي غيرناه بدلا من أن نضع العمليات فوق الـ **(return)** ونخرج بناتج ونضعه بالـ **(return)** هنا وضعنا العملية كاملة بداخل **(return)** وبالتالي خروج النتيجة وإرجاعها مباشرة . (ولكن هذه العملية ليست دائما يمكن إستخدامها) لذلك يفضل إستخدام الصيغة الاولى وبشكل اكثر دقة حسب الحاجة .

عزيزي المتدرب قد يكون الامر لم يتوضح لديك تماما الى الان لاعليك هذا أمرا طبيعي فأنت هنا تتعلم !

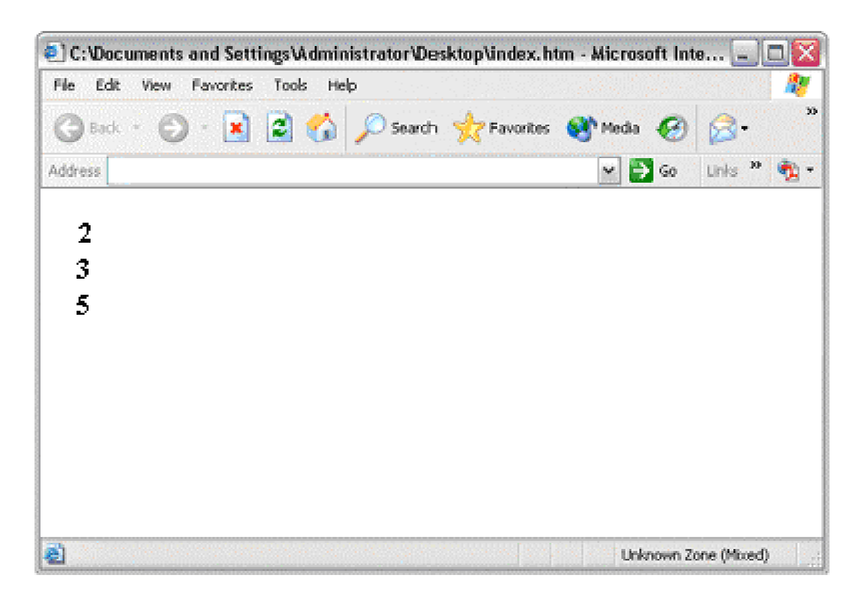
إذا لنأخذ أمثلة تطبيقية فهي التى توضح لنا ما نريده.

المثال الأول :

```
<html >
    <head><title>الجافا سكريبت</title>
      <script type = "text/javascript">
          var  z ;
          var  x=2 ;
          var  y=3 ;
          document.write ( x ) ;
          document.write ( " <br> " + y ) ;
          z = summation ( x , y ) ;
          document.write ( " <br> " + z ) ;

       function   summation ( x , y )
           {
              var  sum = 0 ;
              sum= x + y ;
              return  sum ;
           }
      </script>
    </head>
  <body></body>

</html>
```



لنحلل المثال سويا . . .
قمنا هنا بتعريف ثلاثة متغيرات المتغير الاول وهو **(x)** ووضعنا بداخله القيمة الرقمية الصحيحة **(2)** والمتغير **(y)** ووضعنا بداخله القيمة الرقمية الصحيحة **(3)** والمتغير **(z)** الذي سوف نضع به القيمة التى ترجعها لنا الدالة **(summation)** أي ناتج الجمع **(x+y)** .

ثم قمنا بطباعة قيمة **(x)** من خلال الأمر **(document.write (x))** وبالتالي طباعة رقم **(2)** فهذا الرقم الذي وضعناه بالـ **(x)** .
ثم كتبنا جملة الطباعة الخاصة بطباعة قيمة **(y)** وذلك من خلال الامر :

document.write (" < br > " + y)

ثم قمنا بوضع مايلي :

```
z = summation ( x , y ) ;
```

ماذا نعني بها . . .
نعني بها أن قيمة المتغير **(z)** تساوي القيمة التي سوف ترسلها الدالة **(function)** التي أسمها **(summation)** التي أرسلنا لها قيمتين وهما قيمة **(x)** وقيمة **(y)** لكي ترجع اي ترسل لنا الناتج لنضعه في **(z).** فنحن هنا نحصل على قيمة **(z)** من القيمة التي سترجعها الدالة.
إذا هنا نحن أستدعينا الدالة **(summation)** لكي ترجع لنا ناتج العملية ومن ثم نضعه في **(z)** إذا لنحصل على هذا الناتج.. نذهب الى الدالة المسماة **(summation)** لنذهب لها وسوف نجد بداخلها مايلي :

عرفنا متغير أسميناه **(sum)** وصفرناه " أي جعلنا قيمته صفرا " ووضعنا الصفر فقط لكي نحسب الرقمين بدقة وعدم وجود أي قيمة سابقة للـ **(sum)** فكما ترى عندما تفتح الالة الحاسبة ألا تضع لك صفرا ثم تضغط الرقم **(2)** ثم تضغط **(+)** ثم تضغط رقم **(3)** فيخرج لك الناتج **(5)** أي الصفر تحول الى **(5)** فهنا كأننا نقول أن قيمة الـ **(sum)** قبل العملية كانت صفرا وبعد العملية أصبحت **(5).** لنرجع لموضوعنا اذا ...
الان جعلنا **(sum)** تساوي قيمة الـ **(x)** المرسلة وجمعنا لها قيمة الـ **(y)** المرسلة أيضا وكما تعرف " قيمة **(y)** وقيمة **(x)** سوف تعرفهما الدالة مباشرة من خلال أرسالنا في البداية عند أستدعائها قيمة الـ **(x)** وقيمة **(y)** أي **(2)** و **(3)** "
.
وهنا سوف تتم عملية الجمع أي **(2) + (3)** ويكون الناتج **(5)** الذي سوف نضعه في المتغير **(sum)**
إذا الان أصبحت قيمة **(sum)** هي **(5)** وهي ناتج العملية .

ثم كتبنا **(return)** ووضعنا بها متغير **(sum)** أتعرف عزيزي المتدرب ماذا تعني **(return)**؟
تعني أرجع . . إذا نحن سوف نرجع قيمة **(sum)** والتى هي **(5) الى أين ؟؟؟**
الى المكان **(الام)** الذي أستدعينا فيه الدالة وهو :

```
z = summation ( x , y ) ;
```

إذا هنا سوف يعيد للدالة **(summation)** قيمة **(5)** ثم وضعنا هذه القيمة بالمتغير **(z)** وذلك من خلال :

```
z = summation ( x , y ) ;
```

اي هنا سوف نرسل **(sum)** ليس كمتغير وانما كقيمة اذا سوف يرسل رقم **(5)** وهذا رقم خمسة ماذا يمثل..؟
انه يمثل ناتج الداله ...
(summation (x , y ومنها سوف يتم اسناد القيمة **(5)** الى **(z.) .**

و بعد تنفيذ عملية استدعاء الداله يحق لنا الانتقال الى الخطوه التي تليها في الكود فلا يجوز الذهاب الى الخطوه التي تليها إلا بعد استدعائها وتنفيذها لترجع لنا الناتج ما نفذته وبعد ذلك ننتقل للذي يليها وهي بطباعة قيمة **(z)** من خلال الامر :

```
document.write ( " < br > " + z ) ;
```

فطبع لديناالقيمة (5) أي (2) + (3) = (5) و لاحظ عندما وصلنا الى (return) كنا قد أنتهينا تماما من العمل بداخل الدالة فلقد خرجنا منها وعدنا الى البرنامج " الذي بالاعلى " ولكن معنا قيمة (sum) .

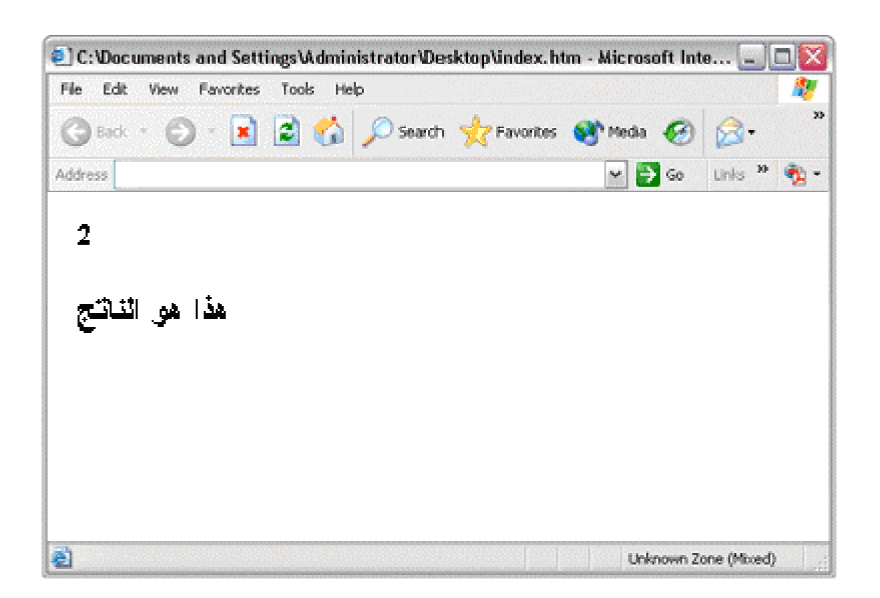
ألم تتساءل عزيزي المتدرب هل يمكن للدالة (function) أن تأتي من غير أن نذكر بها (return) أي أنها لا ترجع أي شئ ؟؟؟.
نعم قد لاتحتوي الدالة (function) على (return) وبما أنها لا تحتوى على (return) إذا هي لن ترجع قيمة ففي هذه الحالة سوف نرى المثال التالي الذي يوضح لنا ...

مثال :

```
<html>
    <head><title >الجافا سكربت</title>
        <script type = "text/javascript">
            var  x = 4 , y = 2  ;
            subtract ( x , y ) ;
            document.write (" <h3>  هذا هو الناتج  </h3> ") ;

           function   subtract ( x , y )
             {
                 var  sub = 0 ;
                 sub = x - y ;
                 document.writeln ( sub ) ;
             }
        </script>
      </head>
    <body></body>

</html>
```



لنحلل المثال سويا :
هنا قمت بتعريف المتغيرات (x) و (y) ووضعت قيمة في كلا منهما ففي الـ (x) وضعت (4) وفي الـ (y) وضعت (2) ثم وضعت اسم الدالة (function) التى أريد أن أستدعيها " أي التى سوف تقوم بعملية الطرح بين الرقمين " وهنا أرسلت للدالة (x) و(y) المحتويان على رقمين أي كأنني أرسلت (4) التى هي (x) و (y) التى هي (2) إذا هنا بما أننا أستدعينا الدالة (function) إذا وجب علي أن أذهب الى هذه الدالة (function) و أقوم بالعمليات التى هي بداخلها ...
لنقوم بالذهاب لها ...

كتبنا أسم الدالة طبعا ووضعنا بين قوسين القيمتين المرسلة لها ثم دخلنا بداخل الداله ثم عرفنا متغير **(sub)** الذي سوف نضع به ناتج عملية الطرح وهي **(x)** ناقصا **(y)** ووضعنا الناتج في **(sub)** من خلال هذه المعادلة :

```
sub = x – y
```

ثم كتبنا أمر طباعة المتغير **(sub)** الذي هو ناتج العملية من خلال الامر :

```
document.write (sub) ;
```

وبالتالي سوف يطبع رقم **(2)** ثم اغلقنا الداله . . .
ثم عدنا مرة أخرى الى مابعد أستدعاء الدالة " أي الى الاعلى " لكننا هنا عدنا ولايوجد بحوزتنا أي قيمة هو مجرد رجوع لأكمال البرنامج فنحن نرى أستدعاء الدالة فتذهب لها لتنفيذها ثم نعود لنكمل بقية البرنامج أي الاوامر التى تلي أستدعاء الدالة وهنا طبعا وضعنا أمر طباعة وهو طباعة الجملة " هذا هو الناتج" من خلال الامر :

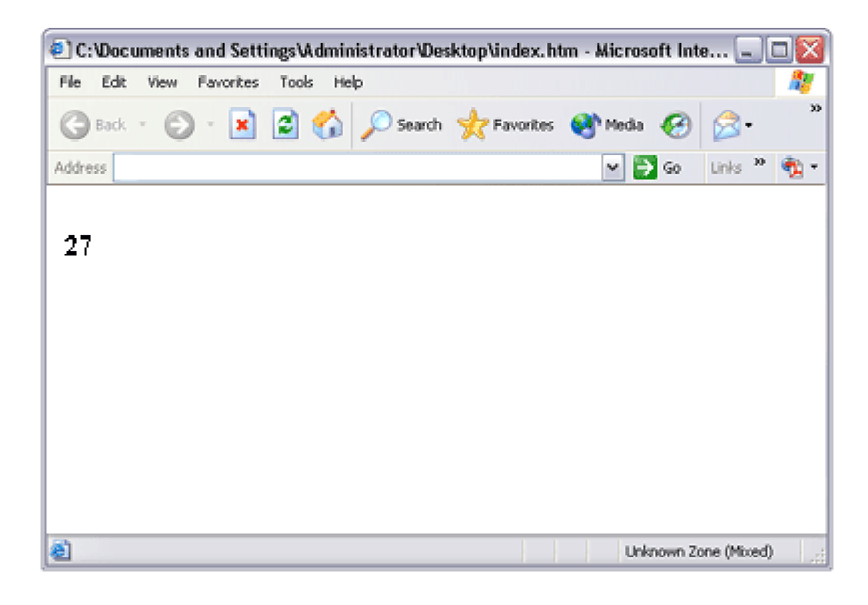
```
document.write ( " <h3>  هذا هو الناتج </h3> " ) ;
```

لاحظ أننا طبعنا الناتج من خلال أمر الطباعة الموجود في الدالة ثم عدنا الى البرنامج فطبعنا الجملة النصية .
ولكن هل من الممكن أن يحتوي برنامجي المكتوب بالجافا سكربت على دوال فقط ... كأن دالة نستدعي دالة ..
نعم ممكن ذلك لنأخذ مثال ونرى :

```
<html>
    <head><title >الجافا سكربت</title>
        <script type = "text/javascript">
          function   output ( )
            {
                var  x = 3 , t ;
                 t = cube ( x ) ;
                document.writeln ( t ) ;
            }

          function cube ( y )
            {
                 return  y*y*y ;
             }
        </script>
     </head>
   <body onload = " output ( ) " ></body>

</html>
```



وسوف أجعل تحليل هذا المثال لك ولكن سوف أوضح فقط بعض الامور التى تساعدك في تحليله... التى أعتقد أنها جعلت في فكرك بعض التساؤلات عندما رأيت كود البرنامج .

١- قد تتساءل لماذا لم نضع متغير بين القوسين عزيزي المتدرب هل هذه الدالة سوف تستقبل متغير " أو حتى قيمة من الخارج " الجواب لا إذا أدعها فارغة ولا أضع بها شيء فهي دالة لاتستقبل .

```
function  output ( )
```

٢- قد تتساءل لماذا كتبنا هنا **(y)** مع أننا عندما أستدعينا هذه الدالة من الدالة **(output)** كتبنا بهذا الشكل :

```
T = cube ( x ) ;
```

```
function  cube ( y )
```

أي وضعنا **(x)** وليس **(y)** عزيزي المتدرب فليس مهما أن تسمي المتغير الذي أرسلته بنفس أسم متغر الدالة المرسل لها فهذا ليس مهما طبعا ليس مهم تسمية المتغير في الدالة المرسل لها أي أقصد :
(function cube (y)) فهنا المهم أننا أرسلنا له **(x)** والتى قيمتها **(3)** .
لاتنزعج تستطيع أن تسمى بدلا من **(y) (x)** أي تصبح **(function cube (x))** فقط ذكرتها من باب العلم بالشئ .

٣- هنا جعلناه يضرب رقم **(3)** بنفسه ثلاثة مرات وإرجاع النتيجة مباشرة . وهذا بدلا من تعريف متغير ونضع النتيجة بداخله ومن ثم نرسل هذا المتغير بـ **(return)** ونرجع القيمة قمنا بوضع العملية الحسابية مباشرة في ال **return** ... " لاحظ أننا ذكرنا سابقا اننا لم نضع في ال **( ) function output** متغير بداخله وهذا لانه لن يستقبل متغير ولنقل بشكل اكثر دقه **parameter** اي اننا لن نمرر عليه قيم.. واذكرها هنا لك لكي لا تقول في نفسك ها نحن عدنا له بقيمة عملية الضرب ثلاث مرات اذا استقبل **!** عزيز المتدرب هنا العملية تمت من خلال **return** وليس ارسال قيم لتمريرها ليتم عليها عملة معينه اي هنا تم استدعاء دالة لتقوم بعملية وترجع لنا قيمه .. تذكر عندما كنا نستدعي الدالة من داخل برنامج من دون ان يكون بادخل دالة اخرى .

لاتنزعج تستطيع أن تكتبه كما تعلمنا سابقا أي :

```
return  y*y*y ;
```

```
var  z ;
z = y*y*y ;
return  z ;
```

فقط أحببت أن أريك طريقة ثانية من باب العلم بالشئ فقط لا غير وقد تلجأ له عند الحاجه .

٤- معلومة مهمة جدا ...
عندما تكتب برنامجك بالجافا سكربت ويكون يحتوي فقط على دوال كما في مثالنا هذا فيجب عليك أن تذكر في الـ **(body)** هذه العبارة :

```
< body onload = " output ( ) " >
```

§ قد تتساءل لماذا وضعنا **(onload)** داخل الـ **(body)** ؟
الجواب لان برنامجنا عبارة عن دوال والدوال لاتعمل الا عند إستدعائها فأعتبر الـ **(onload)** هو إستدعائها " أي تشغيلها " أما لماذا ذكرنا أسم الدالة الاولى **(output ( ))** فهذا بسبب أن هذه الدالة هي الرئيسية فهي التى تستدعي الدالة الثانية . ولكن ليس دائما نضع **(onload)** مثلا في حالة ال **(form)** وسوف نتحدث عن هذا الموضوع لاحقا .

§ المصفوفات (Arrays) :

هنا سوف نتطرق الى موضوع ذا أهمية وهو المصفوفات فبإستعمالها نستطيع توفير الوقت والجهد بالاضافة الى جعل البرنامج أصغر حجما وأسرع تنفيذا .

وتكتب على أكثر من شكل منها :

١- هنا عرفنا متغير هو عبارة عن مصفوفة تحتوي على **(5)** عناصر أما ماقيمة هذه العناصر فسوف نذكرها داخل البرنامج **.**

```
var  a = new Array ( 5 ) ;
```

٢- وهنا عرفنا المتغير على أنه مصفوفة دون تحديد كم عنصرا تحتوي وهنا يجب أن نحدد كم عدد عناصر المصفوفة في داخل البرنامج " اقصد في الـ **(for)** او ما شابه " كأن نذكرها في حلقة التكرار مثلا أما ما قيمة هذه العناصر فسوف نذكرها أيضا في البرنامج **.**

```
var  a = new Array ( ) ;
```

٣- وهنا عرفنا المتغير على أنه مصفوفة تحتوي على ثلاثة عناصر فوضعنا في كل عنصر قيمة رقمية معينة . وعادة تستخدم تعريف المصفوفة بهذا الشكل عندما تكون المصفوفة تحتوي على أرقام .

```
a = [ 3, 14, 6 ] ;
```

٤- هنا عرفنا المتغير **(a)** على أنه مصفوفة تحتوي على ثلاثة عناصر وكانت قيمة كل عنصر مبينة منذ البداية أي **(1,2,3)** أو **("محمود" ، "ثامر" ، "سامي")** أما كيف عرفنا أن المصفوفة تحتوي على ثلاثة عناصر فهو عدد مابداخلها إذا هي مصفوفة عدد عناصرها ثلاث أما ماهي قيمة عناصرها فهي **(1,2,3)** أو **(محمود ، ثامر ، سامي)** . ولاحظ أنه عندما تحتوي المصفوفة على نص " إذا جاز التعبير **(string)** أي أسم او ما شابه يجب وضع الكلمة بين **(")** .

```
a = new Array ( 1, 2, 3 ) ;

a = new Array ( " سامي " , " ثامر " , " محمود " ) ;
```

§ دعنا نتحدث عن المصفوفات بشكل عام :

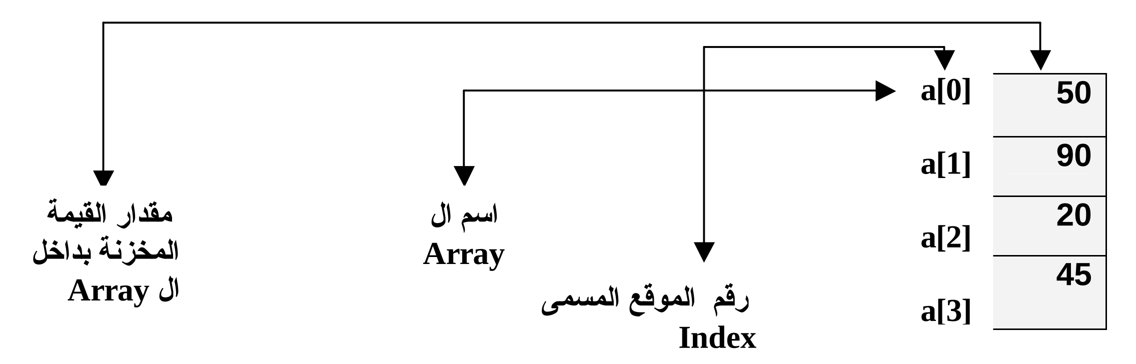
المصفوفة نستطيع تشبيهها بعمارة . العمارة يحتوي كل طابق منها على شقة واحدة . لنفرض أن العمارة تتكون من أربعة طوابق فكان الطابق الأول يسكن به محمد والطابق الثاني أيهم والطابق الثالث أشرف والطابق الرابع عماد . إذا هنا لدينا عمارة تتكون من أربعة طوابق كل طابق يحتوي على شخص فهذا هو الحال بالنسبة للمصفوفة فالعمارة هي أسم المصفوفة . وعدد الطوابق الأربعة هو عدد عناصر المصفوفة والتى هي أربعة والاشخاص الذي كان كل شخص منهم يسكن بطابق هم قيمة كل عنصر في المصفوفة . أي كأننا نقول كل طابق منه يحتوي شخص " فهنا نقول كل عنصر ماذا يحتوي " .

دعنا نطبق هذا المثال بشكل عملي برمجي . .
كلمة عمارة لنختصرها ونسميها **(a)** لنمثل ماقلناه :

```
a = new Array (" محمد "," أيهم "," أشرف "," عماد ") ;
```

إذا هنا الموقع الأول في المصفوفة يساوي محمد والموقع الثاني أيهم والموقع الثالث أشرف والموقع الرابع عماد . ولكن عادة في المصفوفات نبدأ من الصفر وليس من الواحد أي نقول موقع الصفر يساوي محمد و الموقع الاول يساوي أيهم والموقع الثاني يساوي أشرف والموقع الثالث يساوي عماد .
عزيزي المتدرب لاتنزعج من ذلك أعتبر العمارة تتكون من طابق أرضي وطابق أول وطابق ثاني وطابق ثالث إذا محمد سوف يسكن في الطابق الارضي وهو الصفر أي موقع رقم صفر في المصفوفة وأيهم في الطابق الأول أيا الموقع الاول في المصفوفة وأشرف في الطابق الثاني أي الموقع الثاني في المصفوفة وعماد في الطابق الثالث أي الموقع الثالث في المصفوفة .

أما تمثيل المصفوفة بداخل الجهاز اي ال " **Memory** " فيكون بهذا الشكل...
" سوف أضع رسمة توضيحية لتوضح لك كل شئ " :

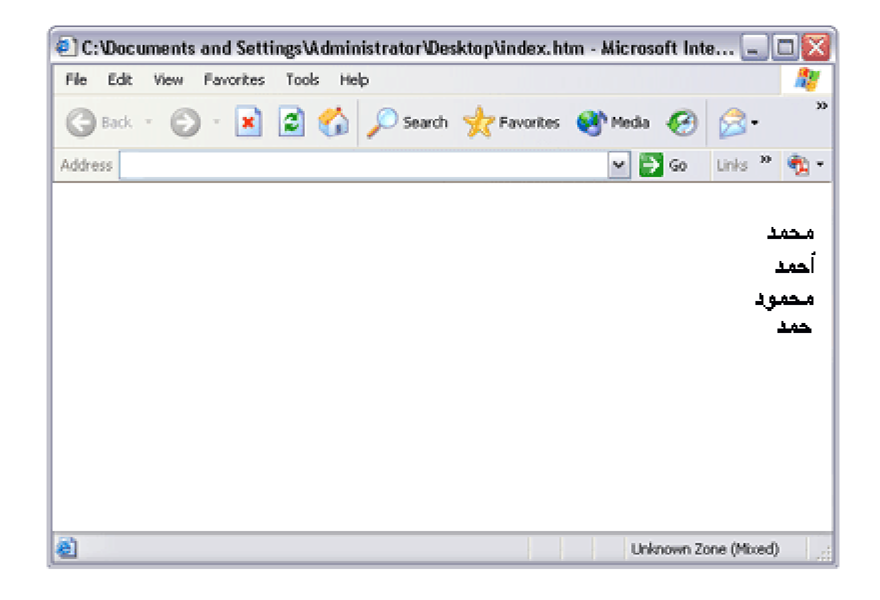


§ قبل أن نبدأ بمثال لابد أن أوضح لك شيئا مهم . . .
بما أن المصفوفة تحتوي على عدة مواقع إذا يجب علي أن أجعل برنامجي يمر على كل موقع إذا هنا بحاجة الى حلقة تكرار لكي استطيع المرور على كل موقع وأطبعه او أنفذه .
أكثرنا الحديث . . لنأخذ مثالا فالمثال التطبيقي والعملي هو الذي يوضح ما بهم لك .

مثال :

```
<html dir="rtl">
    <head><title>الجافا سكربت</title>
        <script type = "text/javascript">
            var  name = new Array (" محمد "," أحمد "," محمود ","حمد") ;
            var  i ;
            for (i = 0 ; i < name.length ; i ++ )
            document.writeln ( name[i] + " <br /> ") ;
        </script>
    </head>
  <body></body>

</html>
```



دعنا نحلل المثال سويا . .

أول شئ عرفنا أسم المصفوفة ثم وضعنا بها أربعة قيم إذا هنا نحن متفقين أننا عرفنا مصفوفة أسمها **(name)** تحتوي على أربعة مواقع كل موقع يحتوي على إسم بداخله. ثم عرفنا المتغير **(i)** الذي سنستخدمه في حلقة التكرار . أستخدمنا في حلقة التكرار هذه العبارة **(name.lengt)** فتعني أن تستمر حلقة التكرار الى أن ينتهي طول المصفوفة أي أن حلقة التكرار سوف تستمر أربعة مرات لأن المصفوفة تحتوي على أربعة مواقع وبما أننا بدأنا حلقة التكرار من صفر أذا سوف يمشي **0 1 2 3** . أذا صيغة طول المصفوفة التى تكتب بالتكرار تكتب كالتالي :

أسم المصفوفة . length

عزيزي المتدرب إذا أنزعجت منها تستطيع أن لاتضعها . تستطيع وضع رقم **(4)** .. بدلا منها فنحن في هذه المصفوفة جعلناها تحتوي على أربعة مواقع . فبالتالي نجعل حلقة التكرار تمر على الاربعة مواقع . قد تتساءل كيف أضع **(4)** فنحن لدينا **(4)** عناصر إذا يجب أن يكون العد الى **(3)** أي **(0 , 1 , 2 , 3)** اى اربعة اسماء فهنا يعتبر العد لاربعة مواقع . . نعم كلامك صحيح **100%** ولكنني في حلقة التكرار بدأنا من **(0)** ثم قلت أصغر من **(4)** أي أن **(4)** ليست داخلة . فلم أقل أصغر أو تساوي **(4)** . كذلك الحال بالنسبة للـ **(name.length)..**

ذكرت داخل جملة الطباعة **(name[i])** واعني بها أنني أريد طباعة مايحتويه كل موقع من مواقع المصفوفة . أي كأنني أقول أريد طباعة أسم كل شخص يسكن في كل طابق من طوابق العمارة . . . تذكر كما قلنا سابقا . إذا كيف أفعل ذلك . . أكتب أسم المصفوفة ثم أفتح القوس **"["** أضع رقم الموقع . . ثم أغلق القوس **"]"** كأن مثلا أقول **(name[0])** فهذا يعني أريد أن أعرف مايحتوي عليه الموقع صفر في المصفوفة وطبعا هو محمد . وهذا ماسوف يحدث بحلقة التكرار فأول مرة سوف ترسل حلقة التكرار قيمة **(i)** صفرا وبما أنني واضع **(name [i])** إذا سوف يكون **(name [0])** أي أطبع مايحتويه الموقع صفر والذي هو محمد ثم ينزل سطرا ونعود مرة ثانية الى حلقة التكرار وتكون قيمة الـ **(i)** قد زيد عليها واحد لتصبح **(1)** فندخل حلقة التكرار وتكون **(name [1])** أي أطبع مايحتويه الموقع رقم **(1)** وهو أحمد . ثم ينزل سطر ويعود مرة ثانية الى حلقة التكرار فتكون قيمة **(i)** قد زادت واحد لتصبح **(2)** فتدخل قيمة **(i)** الى حلقة التكرار فتصبح **(name [2])** أي أطبع قيمة الموقع الثاني التى هي محمود . ثم تعود قيمة **(i)** بعد أن زيدت بواحد أي **(3)** فتدخل حلقة التكرار فتصبح **(name [3])** أي أطبع قيمة الموقع الثالث في المصفوفة هو حمد . ثم تعود قيمة الـ **(i)** بعد أن زيدت بواحد الى حلقة التكرار فلا يتحقق الشرط والذي هو **(i < name.length)** أي أقل من **(4)** فإذا لايتحقق الشرط فنخرج من حلقة التكرار أي نتوقف . وهنا ينتهي البرنامج ويكون قد طبع لنا :

محمد
أحمد
محمود
حمد

§ **ملاحظة مهمة :**
صيغة **(length . اسم المصفوفة)** نستعملها فقط عندما يكون طول المصفوفة معلوم لدينا عند تعريفنا للمصفوفة . وهنا يمكن ان نعرف طولها على شكلين :
مثلا :

```
a = newArry (4) ;
```

مثلا :

```
a = newArry (2,4,6) ;
```

فهنا نعرف ان المصفوفة تتكون من ثلاثة مواقع .

أما متى لا نستخدم **(a.length)** ففي هذه الحالة ....

مثلا :

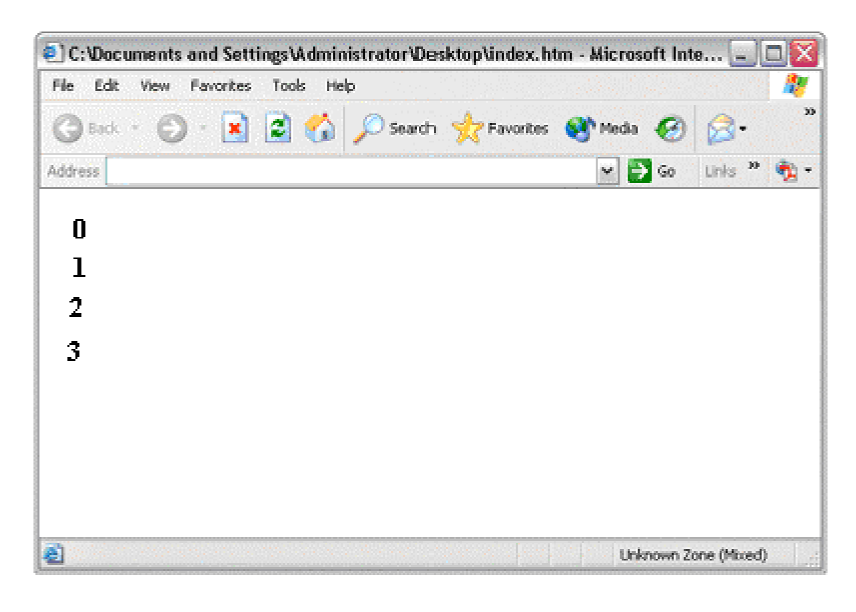
```
a = newArra ( ) ;
```

فهنا نحن اثناء تعريفنا للمصفوفة لم نحدد على كم موقع تحتوي إذا كيف نستخدم **(length.)** فنحن لا نعرف طول المصفوفة أي عدد مواقعها . إذا هنا يجب علينا ان نحدد طول المصفوفة بحلقة التكرار أي بدلا من ان نضع **(a.length)** نضع مثلا رقم **(3)** وهذا يدلنا كم موقع نريد بالمصفوفة .

مثال :

```
<html>
    <head><title>الجافا سكربت</title>
      <script type = "text/javascript">
          var i ;
          var  number=new Array() ;
          for (i=0 ; i<4 ; i++)
            {
              number[i] = i ;
              document.writeln(number[i] + "<br>");
            }
      </script>
    </head>
  <body></body>

</html>
```



تحليل هذا المثال اتركه لك .... ولكن سوف اضع لك ملاحظتين تسهل عليك تحليله ...

- **(i<4)** نعني بها اننا نريد المصفوفة تحتوي على اربعة مواقع " تذكر اننا بدأنا من **(0)** " .
- **(number[i] = i)** نعني بها ان المصفوفة عرفناها في البداية ولم نضع بها قيم . فهنا نضع بها القيم أي كأننا نقول موقع الصفر ضع به **(0)** وموقع الواحد ضع به **(1)** وهكذا . أي :

```
Number [0] = 0
Number [1] = 1
Number [2] = 2
Number [3] = 3
```

اذا نحن هنا قمنا بتعبئة مواقع المصفوفة بقيم ومن ثم طبعناها .

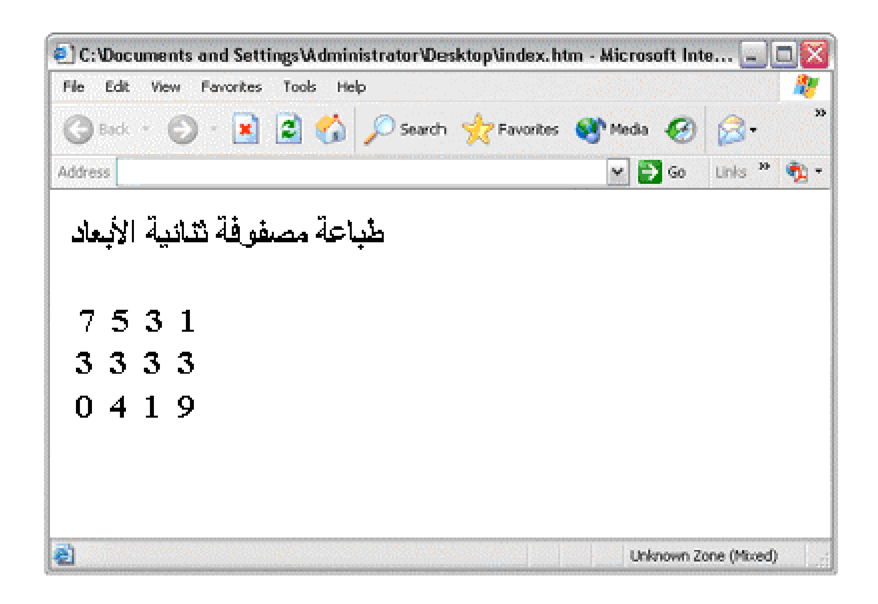
§ عزيزي المتدرب نكون هنا قد انهينا الحديث عن المصفوفات ذات البعد الواحد المسماة :
" One Dimension Array " (1D) .

§ إذا بقي علينا الشكل الاخر للمصفوفة وهي المصفوفة المسماة :
" Tow Dimension Array " (2D) .

لنأخذ مثالا ونوضح ما هي ...

```
<html>
  <head><title>الجافا سكريبت</title>
     <script type = "text/javascript">
       var arr = [ [1,3,5,7] , [3,3,3,3] , [9,1,4,0] ] ;
       document.writeln("<h2>طباعة مصفوفة ثنائية الأبعاد</h2><tt>");
       for (i=0 ;i< arr.length;++i)
          {
             for (j=0 ;j< arr[i].length;++j)
                  document.writeln(arr[i] [j] + " ");
             document.writeln("<br />");
          }
        document.writeln("</tt> ");
     </script>
   </head>
  <body></body>

</html>
```



لنحلل المثال سويا ...

أخي المتدرب شرحي في هذا المثال سوف يكون طويلا وقد أثقل عليك من كثرة ما سوف تقرأ ولكن ذلك من أجل إيصال المعلومة لك بالتفصيل إلى أبعد الحدود فأعذرني فكما هو شاق عليك بالقراءة هو شاق علي بالكتابة والشرح .....

هنا قمنا بتعريف المصفوفة وكانت هذه المصفوفة تحتوي بداخلها على ثلاثة مصفوفات لا تنزعج من ذلك فقد كنا نقول في الماضي ان المصفوفة تحتوي على ثلاثة عناصر وهنا نقول تحتوي المصفوفه على ثلاثة مصفوفات أي عملية تداخلية . الان يجب أن نعرف شيئا ...
هذه المصفوفة التي بداخلها مصفوفات هي (كم عامودا في كل صف) **(rows & columns)** وهنا طبعا مصفوفتنا هذه سوف تتكون من أربعة أعمدة وثلاثة صفوف . إذا أبعادها **(4 x 3)** كيف عرفنا ذلك ؟

لاحظ أن المصفوفة الرئيسية كانت تحتوي على ثلاثة مصفوفات بداخلها إذا المصفوفة الرئيسية تحتوي على ثلاثة صفوف وكان كلا من هذه المصفوفات يحتوي على اربعة عناصر إذا المصفوفة الرئيسية أيضا سوف تتكون من أربعة أعمدة وهي عدد العناصر في المصفوفات التي بداخلها فقد كانت كل مصفوفة من هذه المصفوفات تحتوي على قيم بداخلها وكانت كل واحده تتكون من اربعة مواقع اذا . . .
ثم كتبنا أمر طباعة جملة نصية والتي هي " طباعة مصفوفة ثنائية الأبعاد " ووضعنا هذا الوسم **(<tt>)** الذي سوف أغلقه بعد الانتهاء كليا من البرنامج فقط لكي يطبع لي البرنامج كل الارقام التي تحتويها المصفوفة بشكل الآلة الطابعة وتستطيع ان لا تضعه .
قبل الدخول الى حلقة التكرار أنت تعلم جيدا ان حلقة التكرار في كل مرة ندخل بها داخلها يجب العودة لها أكثر من مرة حتى يفشل الشرط فنخرج منها " أي عدم تحقق الشرط " فماذا لو كان لدينا حلقة تكرار بداخلها حلقة تكرار أخرى في هذه الحالة ندخل في حلقة التكرار الاولى أي الرئيسية فنجد بداخلها حلقة تكرار أخرى وهنا سوف ندخل بداخل هذه الحلقة ونستمر بتنفيذ ما بداخلها والعودة لها حتى يفشل شرطها وبالتالي الخروج منها والعودة الى حلقة التكرار الاولى كدورة ثانية لها فندخل بداخلها فنجد حلقة التكرار الثانية مرة أخرى فنستمر بتنفيذ ما بداخلها أي التكرار حتى يفشل الشرط " أي عدم تحققه " .
ثم نخرج من هذه الحلقة ونعود الى الحلقة الرئيسية مرة أخرى وهكذا حتى يفشل شرط حلقة التكرار الرئيسية وبالتالي عدم الدخول بداخلها وعدم الدخول بحلقة التكرار التي بداخلها " أي الحلقة الثانية " وبالتالي الخروج تماما .

الان لندخل بحلقة التكرار ببرنامجنا ونشرحها ولكن قبل ذلك دعنا نتفق على مسميات مجازية " إذا جاز التعبير " استخدمها فقط من أجل التوضيح لا أكثر :

نسمي حلقة التكرار الاولى التي ببرنامجنا المحتوية على المتغير **(i)** : **بـ " حلقة التكرار الرئيسية "** وحلقة التكرار الثانية أي التي بداخل حلقة التكرار الاولى المحتوية على المتغير **(j) : بـ " حلقة التكرار الداخلية " .**

§ لنبدأ .....

عرفنا هنا ببرنامجنا حلقة التكرار الرئيسية وعرفنا بداخلها المتغير **(i)** وجعلنا حلقة التكرار تبدأ من **(0)** الى طول المصفوفة الرئيسية التي تحتوي بداخلها على المصفوفات . وبما أن لدينا بداخلها ثلاثة مصفوفات إذا سوف تكون قيمة **(arr.length)** هي (3) أي الشرط سوف يكون **(i < 3)** ابتداءا من **(0)** طبعا . أي ان هذه الحلقة سوف تستمر بالتكرار ثلاثة مرات " أي الدخول بداخلها وتنفيذ ما بداخلها ثلاثة مرات " .
لاتنزعج لاني وضعت داخل حلقة التكرار **(var i)** فأنا هنا فقط بدلا من أن أعرف المتغير **(i)** في بداية البرنامجي عرفته بداخل حلقة التكرار . طبعا تستطيع ان تعرفه في بداية البرنامج أو تعرفه داخل حلقة التكرار كما تشاء " فقط أحببت أن اريك عدة طرق لتعريفه ليس عبثا ولكن من باب العلم بالشيء " .
هنا دخلنا حلقة التكرار الرئيسية وقيمة **(i)** هي **(0)** ثم دخلنا حلقة التكرار الداخلية وعرفنا المتغير **(j)** وجعلناه يبدأ من **(0)** ووضعنا شرط ال **(j)** هو **(j<arr[i].length) .** وبما ان قيمة **(i)** هي **(0)** إذا الشرط سوف يكون **(arr[0].length)** ماذا تعني هذه الجمله ؟

تعني ماذا يحتوي الموقع **(0)** في المصفوفة الرئيسية وطبعا يحتوي على مصفوفة تحتوي بداخلها على أربعة عناصر .. أي كأننا نقول بهذه الجملة " كم تحتوي المصفوفة على عناصر بداخلها في الموقع **(0)** من المصفوفة الرئيسية " ...

وبما أنها تحتوي على أربعة عناصر ، إذا هنا ترجمة **(arr[0].length)** تعني **(4)** إذا الشرط هو **(j<4)** إذا سوف ننفذ ما بداخل الحلقة الداخلية أربعة مرات إبتداءا من **(0)** وانتهاءا بـ **(3)** وسنبقى مستمرين ولن نخرج منها حتى ننتهي " أي عدم تحقق الشرط " . فإذا حصل ذلك نخرج منها ونعود مرة أخرى الى حلقة التكرار الرئيسية . إذا لندخل بداخلها ...
فقيمة **(j)** تساوي **(0)** وهو أقل من **(4)** فعلا . فننتقل الى أمر الطباعة وبداخله **(arr[i][j])**

لنحللها سويا ...

الان قيمة **(i)** لدينا هي **(0)** والـ **(j)** ايضا **(0)** إذا **(arr[0][0])** نعني بذلك الموقع **(0)** في المصفوفة والموقع **(0)** في المصفوفة الداخلية فالـ **(i)** عادة بالمصفوفة الرئيسية أي التي تضم بداخلها على ثلاثة مصفوفات أما **(j)** فهي خاصة بالمصفوفة الداخلية أي الموجودة بداخل المصفوفة الرئيسية . إذا توصلنا لنتيجة ...
أن حلقة التكرار الرئيسية هي خاصة بالمصفوفة الرئيسية ومواقعها .
وحلقة التكرار الداخلية خاصة بالمصفوفات الداخلية ومواقعها " أي المصفوفات التي بداخل المصفوفة الرئيسية " .

الان لدينا **(arr[0][0])** اذا يجب الذهاب الى الموقع **(0)** في المصفوفة الرئيسية والموقع **(0)** في المصفوفة الداخلية الموجودة في الموقع **(0)** في المصفوفة الرئيسية فهنا يطبع الرقم **(1)** فنعود مرة أخرى الى حلقة التكرار الداخلية وقيمة **(j)** أصبحت **(1)** " أي بعد الزيادة " فيسألنا الشرط هل **(1)** أقل من **(arr[0].length)** أي هل **(1)** أقل من **(0)** طبعا يتحقق الشرط اذا سوف يدخل الى الحلقة فيطبع **(arr[0][1])** وتعني ذلك اطبع ما يحويه الموقع **(0)** من المصفوفة الرئيسية والموقع **(1)** في المصفوفة الداخلية وبالتالي سوف يطبع الرقم **(3)** وبعد ذلك يعود مرة أخرى الى الحلقة الداخلية وتكون قيمة **(j)** أصبحت **(2)** فيتحقق الشرط لان **(2)** أقل من **(4)** فيطبع قيمة **(arr[0][2])** والتي تعني الموقع **(0)** من المصفوفة الرئيسية والموقع **(3)** في المصفوفة الداخلية ، فيرجع مرة أخرى الى حلقة التكرار الداخلية وقد أصبحت **(j)** تساوي **(3)** بعد الزيادة فيتحقق الشرط فندخل الحلقة فيطبع قيمة **(arr[0][3])** والتي تعني ..

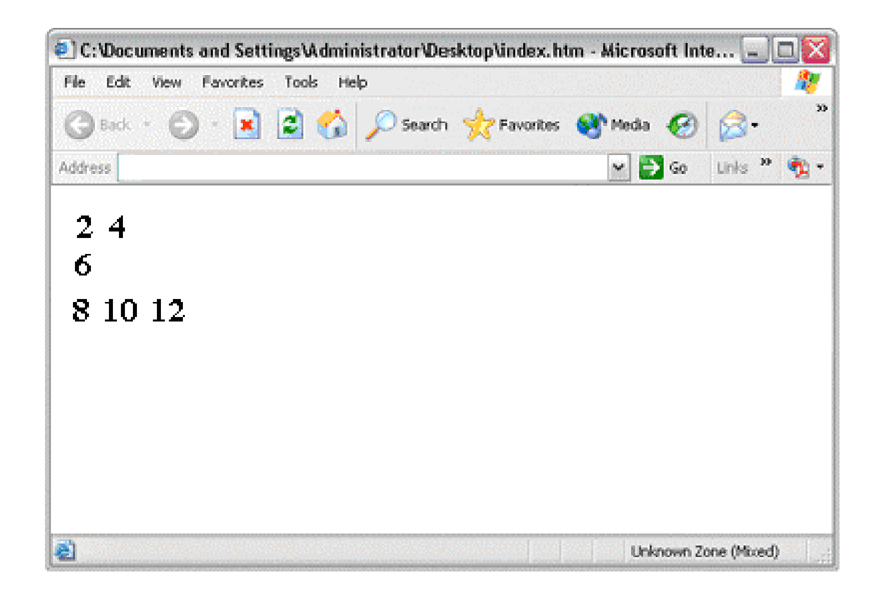
الموقع **(0)** من المصفوفة الرئيسية والموقع **(3)** من المصفوفة الداخلية . أي أنه سيطبع الرقم **(5)** . فنعود مرة أخرى الى الحلقة الداخلية وتكون قيمة **(j)** أصبحت **(4)** بعد الزيادة فيفشل الشرط لان **(4)** ليست أقل من **(4)** . فنخرج من الحلقة الداخلية ونذهب الى أمر طباعة **(
)** أي البدء بسطر جديد ... ونعود لحلقة التكرار الرئيسية ، فتصبح قيمة **(i)** هي **(1) "** أي بعد الزيادة فقد كانت **(0)** وبعد الانتهاء من الدورة الاولى أصبحت **(1)** . فنفحص الشرط الذي هو **"** هل **(1)** أقل من **(arr.length)** أي هل هي أصغر من **(3)** ... فيتحقق وندخل الى داخل الحلقة الداخلية فنبدأ بـ **(j)** تساوي **(0)** ونفحص الشرط والذي هو هل **(j)** أقل من **(arr[i].length)** والذي هو **(arr[1].length)** وتعني طول الموقع **(1)** في المصفوفة الرئيسية وبالتالي فهو **(4)** لان الموقع رقم **(1)** في المصفوفة الرئيسية يحتوي على مصفوفة بداخلها أربعة عناصر . فيصبح الشرط **(4 > 0)** فندخل بداخل الحلقة فنطبع قيمة **(arr[1][0])** أي قيمة الموقع رقم **(1)** في المصفوفة الرئيسية والموقع رقم **(0)** في المصفوفة الداخلية أي سوف يطبع الرقم **(7)** ثم نعود مرة أخرى الى حلقة التكرار الداخلية وتصبح **(j)** قيمتها **(1) "** أي بعد الزيادة **"** فيتحقق الشرط فندخل بحلقة التكرار فنطبع قيمة **(arr[1][1])** والتي تعني الموقع رقم **(1)** في المصفوفة الرئيسية والموقع رقم **(1)** في المصفوفة الداخلية والذي هو **(9)** ونعود مرة أخرى الى حلقة التكرار الداخلية بعد أن اصبحت قيمة **(j)** هي **(2)** فيتحقق الشرط فنطبع قيمة **(arr[1][2])** والتي هي **(11)** فنعود مرة أخرى الى الحلقة الداخلية لتصبح قيمة **(j)** تساوي **(3)** فيتحقق الشرط وندخل بداخلها فيطبع **(arr[1][3])** والتي هي **(13)** فنعود مرة أخرى لحلقة التكرار الداخلية لتصبح **(j)** قيمتها **(4)** فلا يتحقق الشرط " أي يفشل " فنخرج من الحلقة الداخلية ونذهب الى أمر **(< / br>)** أي البدء بسطر جديد ، ثم نعود مرة أخرى الى الحلقة الرئيسية لتصبح قيمة **(i)** هي **(2)** فيتحقق الشرط فندخل بداخل حلقة التكرار فنذهب الى حلقة التكرار الداخلية فنبدأ بـ **(j)** تساوي **(0)** ويصبح الشرط **(j < Arr[2].length)** فيتحقق الشرط لدينا فندخل الى حلقة التكرار الداخلية فنطبع قيمة **(Arr[2][0])** والتى هي **(15)** وهكذا تستمر مع بقاء قيمة **(i)** ثابتة والتى هي **(2)** وقيمة **(j)** التى تزيد بأستمرار حتى يفشل الشرط فنخرج من حلقة التكرار الداخلية فنذهب للأمر **(
)** أي البدء في سطر جديد ثم نعود الى حلقة التكرار الرئيسية لتصبح **(i)** قيمتها **(3) "** أي بعد الزيادة " فيفشل الشرط فالـ **(3)** ليست أقل من **(3)** وبالتالي عدم الدخول لتنفيذ مابداخل حلقة التكرار الرئسيسة " أي بمعنى أننا لن ندخل الى حلقة التكرار الداخلية " إذا هنا نكون قد إنتهينا من الطباعة في حلقات التكرار المتداخلة والصعبة بصراحة . . .

وبعد خروجنا من حلقة التكرار الرئيسية نغلق الوسم **(</tt>)** وهكذا نكون قد أنهينا البرنامج الشاق علينا وعليك . .

مثال:

```
<html>
    <head><title>الجافا سكربت</title>
        <script type = "text/javascript">
            var arr = [ [2,4] ,
                        [6] ,
                        [8,10,12] ] ;
             for (i=0 ; i< arr.length ; ++i)
                 {
                     for (j=0 ; j< arr[i].length ; ++j)
                             document.writeln( arr[i] [j] + " " );
                     document.writeln( "<br />" );
                  }
         </script>
      </head>
   <body></body>

</html>
```



تحليل هذا المثال نتركه لك . . وإذا لم تستطع تحليله جيدا فلا تتردد في مراسلتي لمساعدتك . .

ملاحظة :

```
for (i=0 ; i< أسم المصفوفة.length ; ++i )
            تكافئ
   for (var i in  أسم المصفوفة )

-----------------------------------------------------------------------------

for ( var j=0 ; j< أسم المصفوفة الرئيسية [i]. length ; ++j )
              تكافئ
    for ( var  j  in   أسم المصفوفة الرئيسية [i] )
```

هنا تستطيع أستخدام أي شكل لحلقة التكرار في المصفوفة ولكن أنا أفضل أن تستخدم الشكل الأول . والذي أستخدمه في برامجنا السابقة . فهو أكثر وضوحا لك ولكن هذا رأيي الشخصي وليس بالضرورة أن تعمل به فالامر متروك لك .

§ Math Object :

وهي كائنات او طرق اذا جاز التعبير تقوم بالعمليات الحسابية .. وطريقة استدعائعا او تطبيقها تكون على الصيغة التالية :
اول شيء ذكر اسم ال **(Object)** ثم نقطة **(dot)** ثم بين قوسين يوضع ما هو مراد حسابه **(0)** .. واسم ال **(Object)** هنا هو **(Math)** من عبارة **" mathematical calculation "** اي الطرق الحسابية " اذا جاز التعبير " اذا الصيغة العامه هنا هي :

```
Math .اسم العملية الحسابية ( الرقم او المتغير )
```

مثال :

```
Math.sqrt(9)
```

هنا كاننا نقول نريد الجذر التربيعي للعدد **(9)** اذا الجواب سوف يكون **(3)** وهذا ما سوف يخرجه لنا الصيغة السابقة

ملاحظة :
يجب هنا الالتزام بالحروف من كبيره وصغيره فركز حرف ال **(M)** ياتي بال **(capital letter)** اما البقية تأتي في **(small letter) .... (Math)**

مثال	الوصف	الطريقة
abs(7.2) = 7.2 abs(0.0) = 0.0 abs(-5.6) = 5.6	القيمة المطلقة لـ (x)	abs(x)
ceil(9.2) = 10.0 ceil(-9.8) = -9.0	التقريب لأكبر عدد حقيقي	ceil(x)
cos(0.0) = 1.0	جيب التمام (جتا) لـ (x)	cos(x)
exp(1.0) = 2.71828 exp(2.0) = 7.38906	طريقة الأس (e^x)	exp (x)
floor (9.2) = 9.0 floor (-9.8) = -10.0	التقريب لأصغر عدد حقيقي	floor (x)
log (2.718282) = 1.0 log (7.389056) = 2.0	لوغارثم (x)	log (x)
max (2.3,9.7) = 9.7 max (-2.3,-9.7) = -2.3	أكبر قيمة من (x) و (y)	max (x,y)
min (2.3,9.7) = 2.3 min (-2.3,-9.7) = -9.7	أصغر قيمة من (x) و (y)	min (x,y)
pow(2.0,7.0) = 128.0 pow(9.0,0.5) = 3.0	(x) مرفوع للأس (y) تعني (x^y)	pow(x,y)
round(9.7) = 10 round(9.25) = 9	تقريب (x) لأقرب عدد حقيقي	round(x)
sin(0.0) = 0.0	جيب (جا) لـ (x)	sin(x)
sqrt(900.0) = 30.0 sqrt(9.0) = 3.0	الجذر التربيعي لـ (x)	sqrt(x)
tan(0.0) = 0.0	ظتا (x)	tan(x)

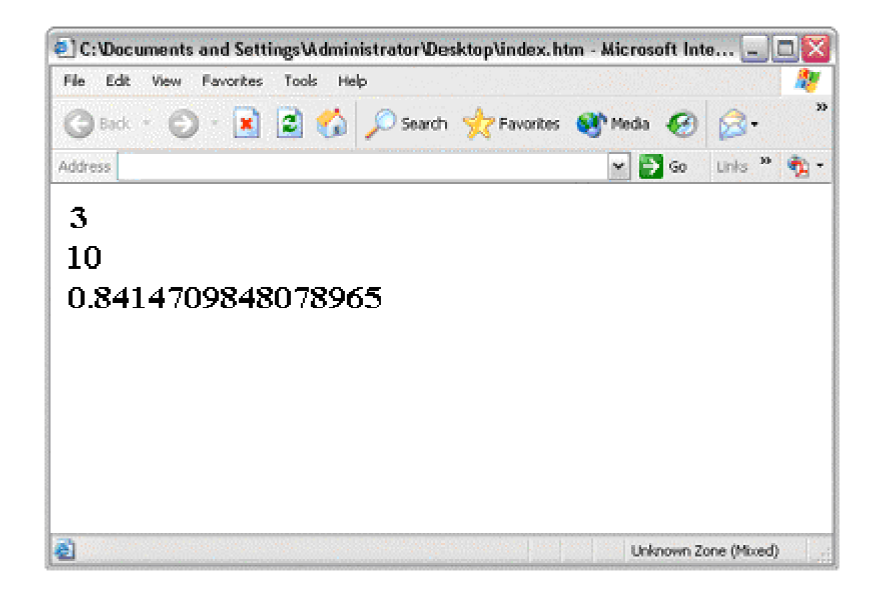
مثال :

```
<html>
   <head><title>الجافا سكربت</title>
      <script type = "text/javascript">

         document.write(Math.sqrt(9));
         document.write("<br />");
         document.write(Math.max(10,2));
         document.write("<br />");
         document.write(Math.sin(1.0));

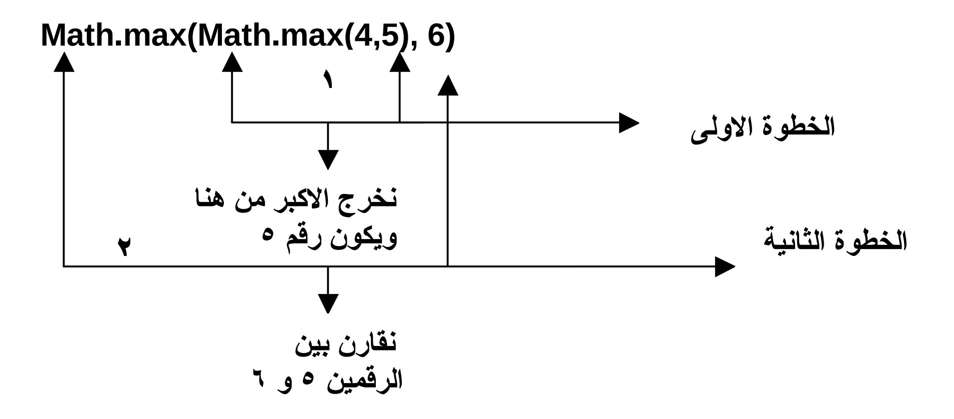
      </script>
   </head>
  <body></body>

</html>
```



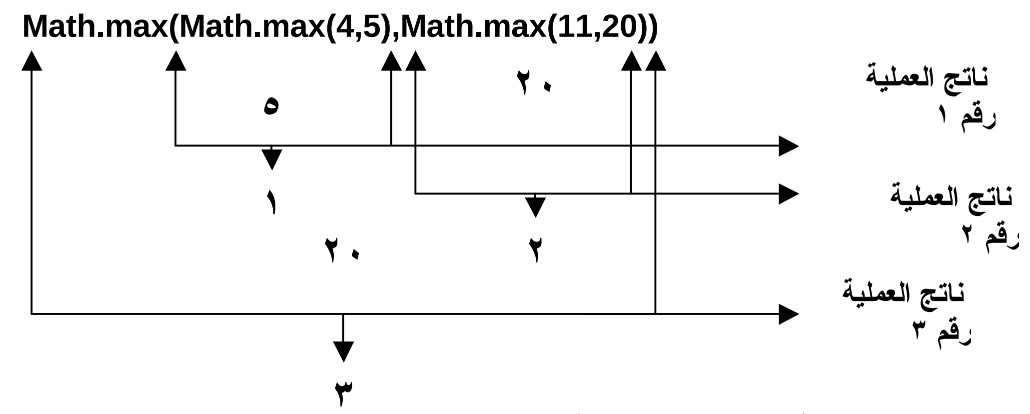
بشكل عام هكذا تكتب ال (**Math Object**) في البرنامج الجافا سكربت ..
فجملة الطباعة الاولى سوف تطبع الجذر التربيعي للعدد (**9**) ثم وضعت ال (**
**) بمفرده في جملة طباعة لكي لا يختلط عليك الامر وذلك لكي ننزل سطرا جديدا ثم تاتي جملة الطباعة الثالثة تطبع لنا الرقم الاكبر بين الرقمين الذين وضعناهما ثم وضعنا جملة الطباعة الثالثة والتي هي نزول سطر جديد " بداية سطر جديد " ثم وضعنا جملة الطباعة الرابعة والتي سوف تطبع لنا جيب الرقم واحد ...

ملاحظة :
لو اردت ان تعمل مقارنه بين ثلاثة ارقام والخروج بالاكبر وقتها تكتب هكذا :



هنا كاننا نقول اخرج الرقم الاكبر في ال **(Math Object)** الذي في الداخل اي بين الرقمين **(4 و 5)** ومن ثم طبق ال **(Math Object)** الذي في الخارج وهو سوف يكون بين الرقمين **(5)** الذي خرج من ال **(Math Object)** التي في الداخل وبين الرقم **(6)** الذي في ال **(Math Object)** في الخارج اي الرئيسية ..

- اذا اردنا ان نخرج الرقم الاكبر بين اربعة ارقام تكتب الصيغة هكذا :
- اي هنا ناخذ كل عملية على حدى وكانها لوحدها ...



§ **تمارين إضافية على الدوال الرياضية :**

الثابت	الوصف	القيمة
Math.E	**قيمة ثابتة**	**تقريبا تساوي 2.718**
Math.LN2	**اللوغاريثم الطبيعي لـ (2)**	**تقريبا تساوي 0.693**
Math.LN10	**اللوغاريثم لطبيعي لـ (10)**	**تقريبا تساوي 2.302**
Math.LOG2E		**تقريبا تساوي 1.442**
Math.LOG10E		**تقريبا تساوي 0.434**
Math.PI		**تقريبا تساوي 3.141592653589793**
Math.SQRT1_2		**تقريبا تساوي 0.707**
Math.SQRT2	**الجذر التربيعي لـ (2.0)**	**تقريبا تساوي 1.414**

§ **: Methods of the String Object**

الطريقة	الوصف
charAt(index)	**يرجع الحرف المحدد بالـ (index)**
charCodeAt(index)	**يرجع شيفرة الحرف المحدد بالـ (index)**
concat(string)	**يدمج النص المحدد في الـ (string) المحدد قبل الامر (concat) مع النص المحدد في الـ (string) الذي بين القوسين اي الذي بعد (concat)**
fromCharCode(value1,value2 ,...)	**يحول الارقام المحددة بين الاقواس الى قيمتها الحرفية**
indexOf(substring,index)	**البحث عن النص المحددة في (substring) من بعد الموقع رقم (index) من النص المحدد في الـ (string)المحدد قبل الأمر. ويرجع أول موقع يحتوي على أول حرف من النص المحدد في (substring) من الـ (string)**
lastIndexOf(substring, index)	**البحث عن النص المحددة في (substring) من قبل الموقع رقم (index) من النص المحدد في الـ**

	(string)المحدد قبل الأمر. ويرجع أول موقع يحتوي على أول حرف من النص المحدد في (substring) من الـ (string)
slice(start,end)	يرجع النص من الموقع (start) الى (end) من الـ (string) المحدد قبل الأمر
split(string)	تقسيم النص المحدد في (string) الى مجموعة كلمات نحن نحدد كيفيتها .
substr(start,length)	يرجع النص المحدد من الموقع (start) وعدد أحرفه (length) من الـ (string) المحدد قبل الأمر .
substring(start,end)	يرجع النص المحدد من الموقع (start) الى الموقع (end)
toLowerCase()	يحول الـ (string) المحدد قبل الأمر الى حروف انجليزية صغيرة .
toUpperCase()	يحول الـ (string) المحدد قبل الأمر الى حروف انجليزية كبيرة .
toString()	يرجع نفس الـ النص المحدد في الـ (string)
valueOf()	يرجع نفس الـ النص المحدد في الـ (string)
anchor(name)	يعمل عمل (<a> name)
blink()	يعمل عمل (<blink>…</blink>)
fixed()	يعمل عمل (<tt>…</tt>) أي الكتابة على شكل آلة طابعة
link(url)	يعمل عمل (<a>…)
strike()	يعمل عمل (< strick>…</strick>) أي يضع خط في وسط الكلام
sub()	يعمل عمل (_…) أي الكتابة تحت السطر
sup()	يعمل عمل ([…]) أي الكتابة فوق السطر

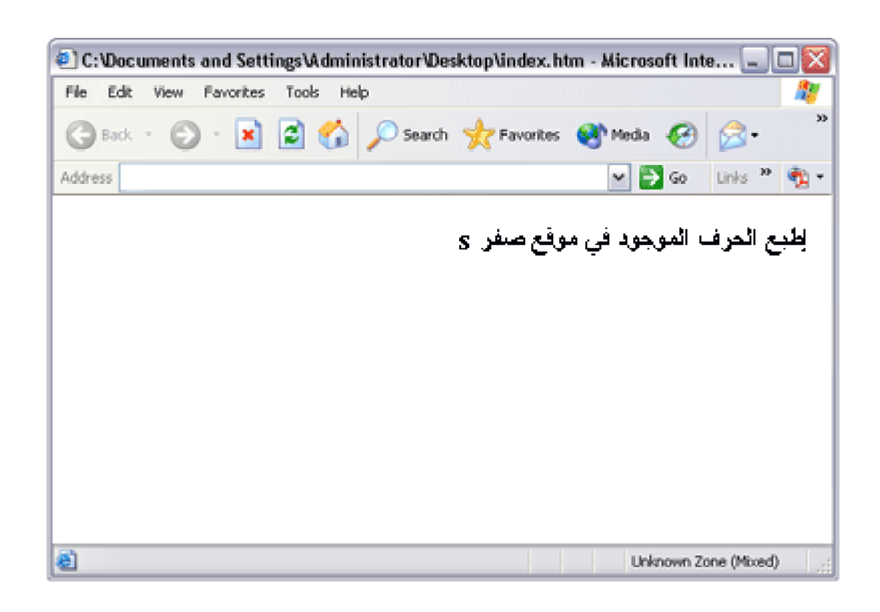
مثال :

```
<html dir="rtl">
     <head><title>الجافا سكربت</title>
          <script type = "text/javascript">

               var  a ="samialr";
               document.write("<p> اطبع الحرف الموجود في الموقع صفر "  + a.charAt(0) );

          </script>
     </head>
   <body></body>

</html>
```



§ Date Object خاصه بالوقت والتاريخ :
وهي كائنات خاصه بالوقت اي تقوم بعمليات ذات علاقه بالوقت من جلب ووضع .

طريقة تعريفه :

```
var a = new Date( )
```

الطريقة	الوصف
getDate()	يعود بالارقام من ١ الى ٣١ حسب اليوم
getUTCDate()	يعود بالارقام من ١ الى ٣١ حسب اليوم
getDay()	يعود بالارقام من 0 " الأحد " الى 6 " السبت "
getUTCDay()	يعود بالارقام من 0 " الأحد " الى 6 " السبت "
getFullYear()	يعود بالسنة من أربعة أرقام
getUTCFullYear()	يعود بالسنة من أربعة أرقام
getHours()	يعود بالساعة من 0 الى 23
getUTSHours()	يعود بالساعة من 0 الى 23
getMilliseconds()	يعود بالميلي ثانية من 0 الى 999
getUTCMilliSecond()	يعود بالميلي ثانية من 0 الى 999
getMinutes()	يعود بالدقيقة من 0 الى 59
getUTCMinutes()	يعود بالدقيقة من 0 الى 59
getMonth()	يعود بالشهر من 0 "يناير" الى 11 " ديسمبر"
getUTCMonth()	يعود بالشهر من 0 "يناير" الى 11 " ديسمبر"
getSeconds()	يعود بالثانية من 0 الى 59
getUTCSeconds()	يعود بالثانية من 0 الى 59
سوف نشرح بعض منها اثناء المثال وهذا لكي تصلك بشكل افضل وما يطبق على البعض يطبق على البقية ..	
setDate(val)	
setUTCDate(val)	
setFullYear(y,m,d)	
setUTCFullYear(y,m,d)	
setHours(h,m,s,ms)	
setUTCHours(h,m,s,ms)	

setMilliSeconds(ms)	
setUTCMilliSeconds(ms)	
setMinutes(m,s,ms)	
setUTCMinutes(m,s,ms)	
setMonth(m,d)	
setUTCMonth(m,d)	
setSeconds(s,ms)	
setUTCSeconds(s,ms)	
setTime(ms)	
toLocaleString()	يعود بالوقت والتاريخ بالتفصيل
toUTCString()	يعود بالوقت والتاريخ بالتفصيل

لنتعرف كيف نتعامل مع ماذكرناه في الجدول :

ولكن قبل ذلك لنأخذ فكرة سريعة عن ماهو المتغير النصي التالي **(string) .**

صيغته :

```
var a = " thamer " ;
```

أي تعرف متغير **(a)** ونضع بداخله نص ولكي تخبر ان هذا نص نضعه بين **(") .**

ملاحظات على هذا المتغير :

١- أي شي يسند للمتغير ويكون بين **(")** سوف نأخذه على أساس أنه نص حتى لو كان رقم .

مثال :

```
var a = " 123 " ;
```

هذا لانأخذه كقيمة رقمية إطلاقا بل نص ليس له قيمة رقمية .

مثال آخر :

```
var a = "ab12 " ;
```

هذا عبارة عن نص كيف عرفنا ذلك لان الرقم جاء بين **(")** حتى الاحرف .

٢- العلامات او الرموز إذا جائت في ال **(string)** تعتبر نص وأيضا الفراغ يعتبر حرف إذا جاز التعبير .

مثال :

```
var a = "ab c12@   " ;
```

هذى المسافة تعتبر حرف **(" @)** أي أنها تأخذ موقع في ال **(Memory)** هو موقع السادس إبتدائا من الصفر **.**

إذا لننتقل الأن الى كيفية التعامل مع ماذكرناه بالجدول .

صيغة كتابتها :

بما أننا نتحدث عن طرق تطبق على الكائنات النصية إذا يجب أن نكون لدينا متغير نصي أي **(string)** لنجري عليه مانريد من الطرق سالفة الذكر في الجدول إذا :

١- نعرف أول شي متغير نصي **(string).**

```
var a =  " Javascript " ;
```

٢- الأ ن نطبق عليه ال **(Methods)** كيف ذلك :

هنا تكون الصيغة التى تريد أن تطبقها على النص .

مثال :

(ذكر الفهرس أو المتغير أو النص أو القيمة) أسم الطريقة . أسم المتغير النصي (string).

لنأخذ شكلان ونرى :

١- نعرف متغير نصي

```
var a = "Javascript " ;
```

٢- نضيف الـ **(Methods of the string object)**

```
a . substring(0,4)
```

أخبرناه أننا نريد أن يطبع لنا أحرف من الموقع صفر من المتغير النصي **(a)** بطول **(4)** أي أطبع الحروف في الموقع **(0 و 1 و 2 و 3)** إذا سوف يطبع خمسة أحرف... وهي **(Java).**

- **لنأخذ الشكل الاخر الان :**

١- نعرف متغير نصي

```
var a = "Javascript " ;
```

٢- نضيف الـ **(Methods of the string object)**

```
a . toUpperCase(0,4)
```

وفي هذه الحالة سوف يطبع لنا كلمة **(javascript)** ولكن بعد تحويل احرفها الى الأحرف الانجليزية الكبيره اذا سوف تظهر لنا بعد طباعتها بالشكل التالي **(JAVASCRIPT)**

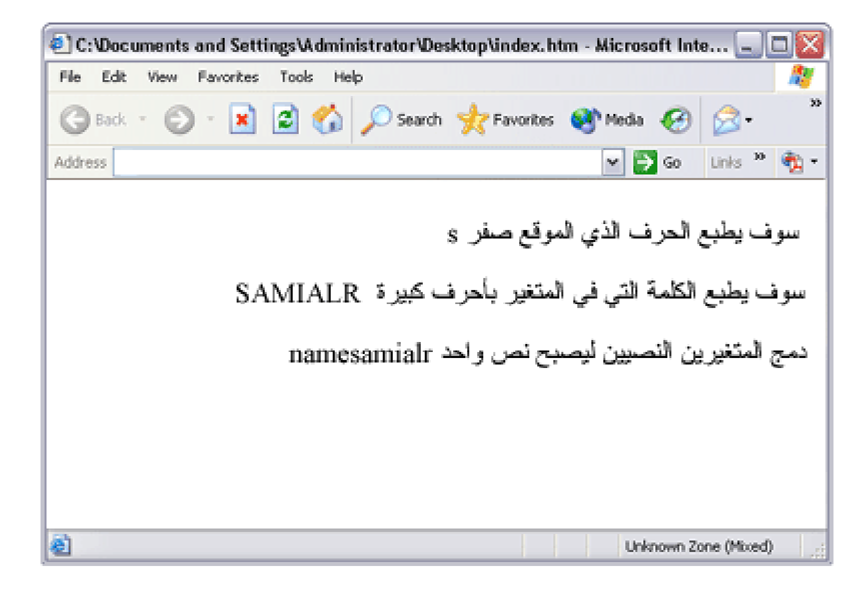
قد تتسائل لماذ لم نضع بعد **(toUpperCase)** قيمة بداخل القوسين عزيزي المتدرب في البداية أخبرتك أنك تضع إذا تطلب ذلك .. ولكن يوجد طرق لايستلزم وجود قيمة أي عليك الالتزام بالجدول.. منها مثلا :

نريد فقط أن نجعل النص يبدو بالحروف الكبيرة إذا كل الذي علينا ذكره أسم المتغير النصي ونقطة ثم كلمة **(toUpperCase)** إذا في المثال السابق بما أننا نريد أن نقول أننا نريد أن نجعل المتغير النصي **(a)** يطبع ولكن ليس بكل أحرفه بل تريد أن يطبع لنا أربع أحرف بداية من الموقع صفر إذا هنا نحن بحاجة الى أسم

المتغير النصي وثاني شي من أي موقع يبدأ العد إذا هنا يجب أن نذكر صفر ثم يجب أن نكتب كم طول الذي نريده من الموقع صفر إذا هنا نذكر **(4)** أي أربع خطوات بداية من الصفر .

```
<html dir="rtl">
      <head><title>الجافا سكربت</title>
          <script type = "text/javascript">
                var a = " samialr ";
                var b = " name ";
                document.writeln("<p> سوف يطبع الحرف الذي في الموقع صفر" + a.charAt (0));
                document.writeln("<p>سوف يطبع الكلمة التي في المتغير بأحرف كبيرة " + a.toUpperCase () );
                document.writeln("<p>  دمج المتغيرين النصيين ليصبح نص واحد " + b.concat (a) );
          </script>
       </head>
    <body></body>

</html>
```



في هذا المثال لن نحتاج لنشرحه لك لان شرح كل امر او خطوه موجود داخل جملة الطباعة ما عليك سوى قراءة المثال وتطبيقه وفهم ما تحتويه جمل الطباعة من اوامر وشرح .

- كيفية التعامل مع **(Date object)** أي كائن التاريخ والوقت إذا جاز التعبير . هنا تخصص الكائن **(Date)** ووضعه في متغير معين ومن ثم تطبيقه على المتصفح بخاصية أو بطريقة من الخواص التي ذكرناه في الجدول،

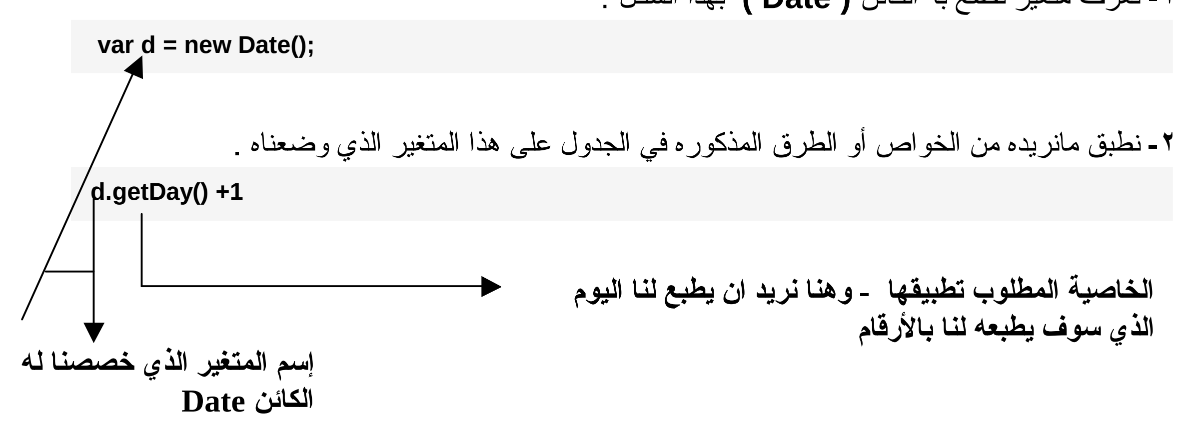
- **إذا كيفية كتابتها :**

١- نعرف متغير نضع به الكائن **(Date)** بهذا الشكل :

```
var d = new Date();
```

٢- نطبق ماتريده من الخواص أو الطرق المذكوره في الجدول على هذا المتغير الذي وضعناه .

```
d.getDay() +1
```



ملاحظة :

وضعنا **(1+)** لانه يطبع لنا اليوم من ٠ الى ٦ ولكننا نريده من ١ الى ٧
ولكن الذي طبقنا عليه اي الكائن هو الخاص بالطرق **(get)** مثل **(getDate())** او **(getMonth())** وهكذا ...
ولكن ماذا عن **(set)** مثل **(setDate(value))** او **(setMonth(m,d))** ... الخ

أقصد بها القيمة التي سوف تحددها انت لتمثل الشهر

أقصد بها قيمة انت تحددها لتمثل اليوم

وهكذا إذا هنا قد نحتاج ان نخبر المتصفح بان يحضر لنا الشهر من الجهاز ويطبعه لنا مثل **(getMonth)** هذا يعني أن نقول للمتصفح أطبع لنا الشهر بعبارة أخرى **(get)** كأننا نقول أحضر الشهر من الجهاز الحاسوب وأطبعه لنا .

أما لو أردنا نحن أن نضع الشهر الذي نريده وليس الموجود بجهاز الحاسوب عندها نكتب **(setMonth(2,1))** وخاصية أو طريقة **(setMonth(m,d))** نضع بداخلها قيمتين كما عرفناها في الجدول ... قيمة لـ **(m)** وتعني قيمة رقم الشهر المطلوب وقيمة لـ **(d)** وتعني لنا اليوم مثل لو وضعنا **(1)** نعني به يوم الأحد وشريطة أن نضع **(1+)** الخاصية لاننا ذكرنا سابقا أنه دائما في الكائن أو التاريخ يبدأ من الصفر و الصفر هو الذي يمثل يوم الاحد . إذا لاحظت أنه عندما نكتب **(get)** كنا نقول أحضر كذا وأطبعه كما هو وعندما نقول **(set)** كأننا نقول ضع كذا وكذا ولكن كما نحدده لك نحن وليس كما هو موجود في الجهاز او الواقع الافتراضي .

ملاحظات :

١- دائما كل الخواص أو الطرق تبدأ من الصفر مثلا :
0 ß يمثل يوم الاحد ، 1 ß يمثل يوم الاثنين ، 2 ß يمثل يوم الثلاثاء ، وهكذا ونفس الشي مع الاشهر.

٢- هنا يطبع لنا دائما الأرقام التي تمثل الاشهر والايام وإذا أردنا أن يطبعها لنا كأسماء نضع الارقام في **(switch)** ونجعله يطبعها لنا كأسماء وسوف نأخذ مثال ونرى لاحقا .

```
<html>
    <head><title>الجافا سكربت</title>
        <script type = "text/javascript">

            var d = new Date() ;
            document.writeln("<p>طباعة رقم اليوم من الشهر <p>"  + d.getDate());
            document.writeln("<p>طباعة رقم الشهر <p>" + d.getMonth());
            document.writeln("<p> طباعة السنة لنا بأربع أرقام <p>" + d.getFullYear());

        </script>
    </head>
  <body></body>

</html>
```

طباعة رقم اليوم من الشهر

2

طباعة رقم الشهر

1

طباعة السنة لنا بأربع أرقام

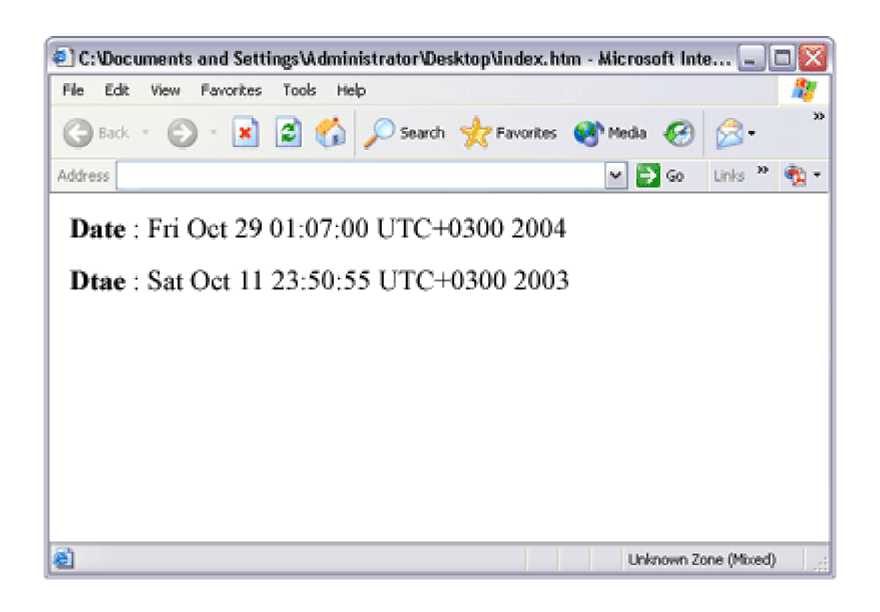
2004

لنأخذ مثالا أخر ونحلله :

```
<html >
      <head><title>الجافا سكربت</title>
           <script type = "text/javascript">
             var c = new Date (2004 , 9 ,29 , 1 , 7, 0 );
             document.writeln("<b>Date :</b> + " c + "<br>");
               c.setDate(11);
               c.setMonth(9);
               c.setHours(23);
               c.setFullYear(2003);
               c.setMinutes(50);
               c.setSeconds(55);
             document.writeln("<b>Dtae : </b>" + c);

          </script>
       </head>
   <body></body>

</html>
```



لنحلل المثال سويا ونرى :

أول شي عرفنا متغير هو المتغير **(c)** ووضعنا بداخله الكائن **(Date)** بشكل **Array** إذا جاز التعبير أي أننا وضعنا بداخله قيم سوف تسند إلى أمور خاصة بالتاريخ والوقت .

```
newDate (2004 , 9 , 29 , 1 , 7, 0 );
```

وبهذا نكون عبرنا عن السنة بالقيمة الاولى وهي **2004** وعبرنا عن الشهر بالقيمة الثانية وهي **9** وعن اليوم بالقيمة الثالثة وهي **29** وعن الساعة بالقيمة الرابعة وهي **1** وعن الدقيقة بالقيمة الخامسة وهي **7** وعن الثواني بالقيمة السادسة وهي **0 .** إذا هنا سوف بتعرف على القيم هكذا : سنة , شهر , يوم , ساعة , دقيقة , ثانية .

ثم كتبنا جملة الطباعة :

```
document.writeln("<b>Date : </b>" + C);
```

وبها نكون قد طبعنا المتغير **(c)** أي بما يحتويه... ومنها سوف يطبع لنا التاريخ والوقت الذي حددناه في بداية برنامجنا عند تعريفنا للمتغير **(c).** ولكن لنقل أننا أردنا أن نغير القيم التي عرفناها في بداية برنامجنا للمتغير **(c)** أي الوقت والتاريخ .لنرى كيف :

```
c.setDate(11);
c.setMonth(9);
c.setHours(23);
c.setFullYear(2003);
c.setMinuites(50);
c.setSeconds(55);
```

بهذه الطريقة نكون كأننا غيرنا كل قيمة في المتغير **(c)** الى هذه القيم التى ذكرناها وهو سوف يغيرها مباشرتا فلا عليك سوي كتابة أسم المتغير ثم نقطة ثم **(set)** وبجانبها الشي الذي نريد تغيره سواءا يوم **(Date)** أو سنة **(FullYear)** وهكذا . ثم الان كتبنا جملة الطباعة وأخبرنا البرنامج أن يطبع هنا المتغير **(c)** مرة أخرى.

```
document.writeln("<b>Dtae : </b>" + c);
```

ولكن هنا سوف يطبع المتغير الذى سوف يعبر لنا عن التاريخ والوقت بالقيم الجديدة التي حددناها قبلها .

- عزيزي المتدرب لقد وضعت لك في هذا البرنامج شكلين لتحديد الوقت والتاريخ بنفسك واحدة محددة من بداية برنامجنا من خلال تعريفنا للمتغير والثانية عندما أردنا تغيير القيم التى يحتويها المتغير وهذا لكي تعم الفائدة وتعرف أبعاد الخاصية .

إذا في النهاية نستنتج شي واحد وهو أن **(get)** عند إستخدامها في التاريخ والوفت تحدد لنا القيم نفسها من الجهاز أما الـ **(set)** نحن من نحدد ونضعهما يدويا.

مثال لنرى كيفية تحويل أرقام الايام الى اسم اليوم كتابتا :

```
<html dir="rtl">
     <head><title>الجافا سكريبت</title>
         <script type = "text/javascript">
                 var d = new Date() ;
                 switch( d.getDay()+1 )
                   {
                       case  1 :
                          document.writeln("يوم الأحد ");
                          break ;

                       case  2 :
                          document.writeln("يوم الأثنين ");
                          break ;

                       case  3 :
                          document.writeln("يوم الثلاثاء ");
                          break ;

                       case  4 :
                          document.writeln("يوم الأربعاء ");
                          break ;

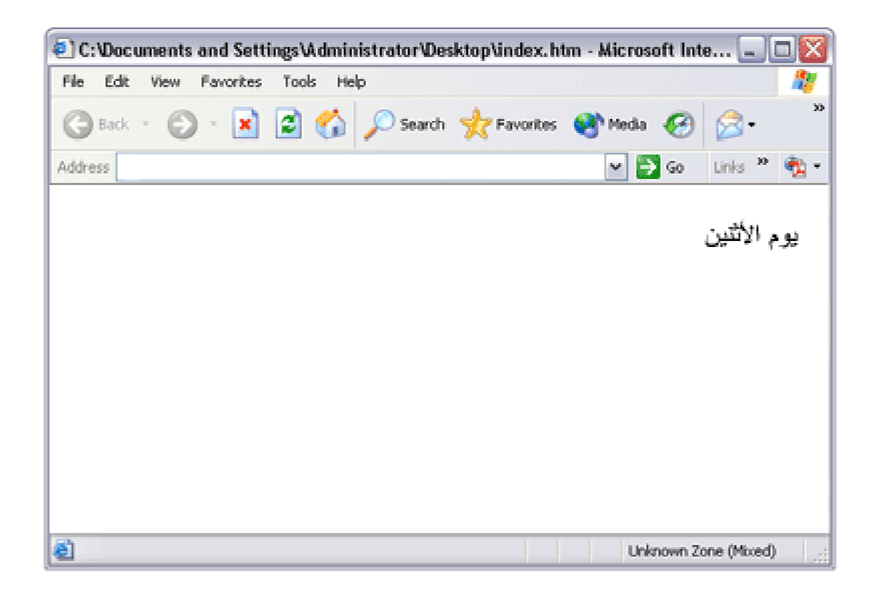
                       case  5 :
                          document.writeln("يوم الخميس ");
                          break ;

                       case  6 :
                          document.writeln("يوم الجمعة ");
                          break ;

                       case  7 :
                          document.writeln("يوم السبت ");
                          break ;
                 }

         </script>
     </head>
   <body></body>

</html>
```



المثال لا يحتاج الى تحليل وذلك لاننا قمنا بشرح ال **(switch)** سابقا في الدرس الخاص به ولكن سوف اضع بعض النقاط الرئيسية التي بها تلخيص لشرح هذا البرنامج .

- عرفنا متغير ووضعنا به الكائن **(Date)** بهذا الشكل :

```
var d = new Date();
```

- وضعنا بداخل الـ **(switch)** كود جلب اليوم بالارقام **(switch(d.getDay()+1))** وبعد ذلك سوف يقارن الرقم الذي سوف يسند الى الـ **(switch)** الى جمل الطباعة لكي يطبع لنا اليوم باللغة العربية اي باحرف اللغة العربية وكان بإمكاننا ان نعرف متغير من ثم نسند له كود جلب اليوم بالارقام ومن ثم نضع هذا المتغير في الـ **(switch)** بهذه الطريقة

```
var d = new Date();
var n = d.gateDay();

switch ( n )
   {
```

ولكنني اختصرت الطريق ووضعت كود جلب اليوم مباشرة في الـ **(switch)** .

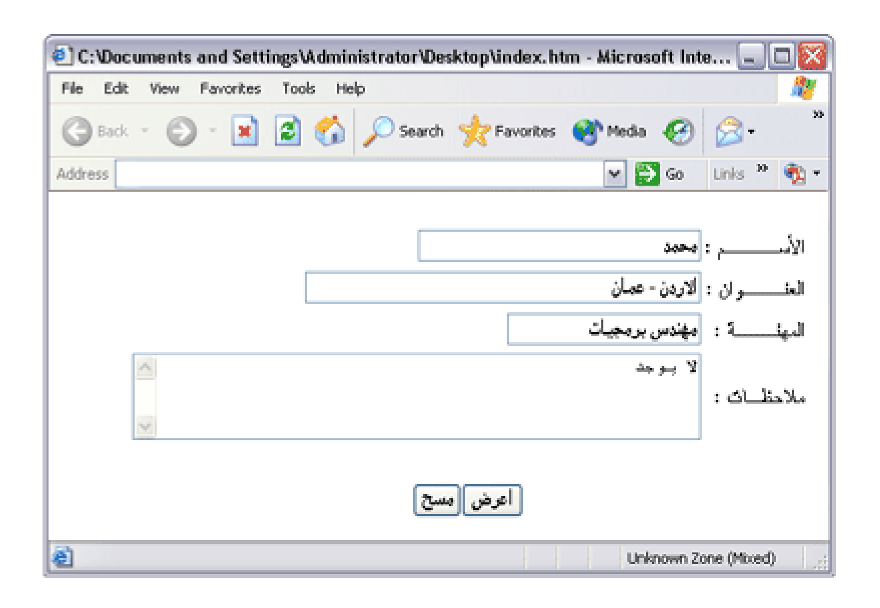
§ **معلومات تطبيقية :**

§ نأتي هنا الى ذكر آخرموضوعين في الكتاب وهما كيفية التعامل بالجافا سكربت مع النماذج **(forms)** والشيء الآخر هو ظهور الكتابة ضمن شريط **(status)** لكي لا أطيل عليك عزيزي المستخدم فأنا أعلم أن الكتاب أثقل عليك من كثر مايحتوي على شرح ولهذا سوف أذكر مثال ونحلله مباشرة فكل شيء يفهم من خلال التطبيق ..

مثال :

وهو تعبئة نموذج من قبل المستخدم وبعد التعبئة يضغط على أيقونة موافق فيطبع بياناته في صفحة الأنترنت

```
<html dir="rtl">
      <head><title> java script </title>
          <script type="text/javascript">
               function information()
                  {
                     var name=info.name.value;
                     var job=info.job.value;
                     var add=info.address.value;
                     var comm=info.command.value;
                     document.writeln("<table cellspasing=\"0\"
                     cellpadding=\"5\" border=\"1\" width=\"80%\"
                     align=\"center\" dir=\"rtl\" bgcolor=\"#eaeaea\"
                     bordercolor=\"#000000\">");
                     document.writeln("<caption><h3>
                     المعلومات التي سجلتها لدينا </h3></caption>");
                     document.writeln("<tbody>");
                     document.writeln("<tr><td width=\"30\"><b> الأســـم  :
                     </b></td><td>" + name + "</td></tr>");
                     document.writeln("<tr><td width=\"30\"><b> المهنـــة
                     :</b></td><td> " + job + "</td></tr>");
                     document.writeln("<tr><td width=\"30\">
                     <b>العنــوان :</b></td><td> " + add + "</td></tr>");
                     document.writeln("<tr><td width=\"30\">
                     <b>ملاحظـات :</b></td><td> " + comm + "</td></tr>");
                     document.writeln("</tbody>");
                     document.writeln("</table>");
                 }
          </script>
     </head>
 <body >
    <form name="info" action="">
    <table cellspacing="0" cellpadding="5" border="0" width="80%" align="center">
      <tbody>
        <tr>
            <td>الأســـم  : </td>
            <td><input type="text" name="name" value="" size="31"></td>
        </tr>
        <tr>
            <td>العنـــوان :</td>
            <td> <input type="text" name="address" value=""   size="45"></td>
        </tr>
        <tr>
            <td>المهنـــة :</td>
            <td><input type="text" name="job" value="" size="20"></td>
        </tr>
        <tr>
            <td>ملاحظـات :</td>
            <td><textarea name="command" rows="4" cols="50"></textarea></td>
        </tr>
        <tr>
             <td colspan="2" align="center">
                </br>
                <input type="button" name="show" value="اعرض"
                onclick="information()"><input type="reset" name="del" value="مسح" >
             </td>
          </tr>
       </tbody> </table></body></html>
```





عزيزي المتدرب لا تنزعج من طول الكود فالكود ليس بطوله بل بأوامره المستخدمه وتكنيكه ومن هنا لا اريد ان احلل الكود لانه لا يحتاج تحليل فلا عليك من رهبة كثرته وطوله ومن هنا سوف اشرح لك الامور التي بها اختصار الشرح بالكود ..

اولا هذا الكود ليس به امور كثيره او اوامر كثيرا بل هو بسيط الى ابعد الحدود لننظر كيف ذلك :

لقد عرفت المتغيرات بالشكل التالي :

```
var name=info.name.value ;
var job=info.job.value ;
var add=info.address.value ;
```

هنا سوف اتجاوز بشرحي قليلا عن النمط البرمجي البحت لكي تصل المعلومه اكثر اذا سمحت لي ..

هنا عرفنا المتغيرات بالشكل الذي نعرفه مثل **(name)** و **(job)** و **(add)**
وتذكر أخي المتدرب اننا كنا نعرف في الماضي... اي في المصفوفات متغير مثلا **(a)** ويكون هو عباره عن مصفوفه وكانت تكتب بشكل التالي :

```
var a = new Array( )  ;
```

ومنها كنا نعرف ان هذا المتغير هو عباره عن مصفوفه ..
وهنا نفس الشيء قلنا مثلا المتغير **(name)** هو عباره عن قيمه سوف يدخلها المستخدم في النموذج **(form)** كما كنا نقول ان هذا المتغير **(a)** هو عباره عن مصفوفه .. لهذا كتبنا المتغير **(name)** بهذا الشكل :

```
var name=info.name.value ;
```

اي هنا نقول ان المتغير **(name)** هو عباره عن قيمه سوف تكون مدخله من قبل المستخدم في حقل من حقول النموذج اي ال **(form)** والذي يحمل اسم **(name)** لكي تخزن بداخلها ..
اذا هنا نأتي للتساؤلك عزيزي المتدرب ... ماذا يعني الكود التالي :

```
info.name.value ;
```

لنفسرها سويا :

§ **(info)** ونعني بها اسم النموذج الذي سوف يدخل به المستخدم القيمه والذي يكتب في ال **(form)**

```
< form name="info" action=" " >
```

فكما تعرف في لغة **(html)** النصيه يجب ان يحمل النموذج اسم اذا هنا يجب ان نضع نفس الاسم الذي حددناه ..للمتغير **(name)** وهذا لكي نخبر ان هذا المتغير متعلق بذلك النموذج ..

§ **(name)** ونعني بها الاسم الذي سوف نضعه بالحقل الذي سوف يدخل به المستخدم القيمه ومن ثم تخزن في المتغير **(name)** والتي تكون في النموذج بهذا الشكل :

```
< input type="text"  name="name"  value="" size="3" >
```

وهذا لكي نميز ان الحقل هذا الذي في ال **(input)** والذي سوف يدخله المستخدم سوف يخزن في المتغير الذي اسمه **(name)** اي انه خاص به .

§ **(value)** ونعني به ان هذه القيمه سوف يدخلها المستخدم لذلك هي غير معلومه الى الان ..

وبهذا تكون الصيغة للمتغير والقيمة التي سوف تخزن بها كتالي :

```
var name=info.name.value ;
```

إذا عرفنا لماذا نضع المتغير ونساويه بهذا الشكل كما في المصفوفه ولكن باختلاف الوظيفه وطريقة الكتابه ..
وما ينطبق على المتغير **(name)** ينطبق على المتغير **(job)** و **(add)**

§ قبل ان ننتقل الى بقية الكود لنتسأل هل المسميات هذه ثابته ويجب ذكرها دائما ..؟
الجواب : لا

فقد نأتي ونسمي المتغير ونكتبه بهذا الشكل. لنقل مثلا نريد ان نسمي متغر بإسم **(a)** فرضا. اذا نكتبه كتالي :

```
var a=myform.enter1.value ;
```

إذا بما اننا غيرنا المسميات يجب ان نثبت بها على النموذج اي ال **(form)** لكي يعرف ان هذا المتغير خاص به اذا التغيرات كتالي سوف تصبح ..

تعريف المتغير :

```
var a=myform.enter1.value ;
```

عند كتابتنا لكود النموذج نذهب و نكتب اسم ال **(form)** بهذا الاسم وهو الذي اخترناه من بداية برنامجنا " **myform** " كتالي :

```
< form name="myform" action=" " >
```

والحقل الذي سوف يدخل من خلاله المستخدم قيمة المتغير التي سوف تخزن به سوف يأخذ اسم " **enter1** " كتالي :

```
< input type="text" name="enter1" value="" size="31" >
```

§ **اذا نستنتج التالي :**
ان المسميات نحن من نحددها ولكن اذا حددناها منذ بداية البرنامج علينا الالتزام بها الى اخر البرنامج دون تغير ..

نأتي الان الى الشيء الاخر وهو الكود التالي وماذا نعني به :

```
<input type="button" name="show" value="اعرض" onclick="information()">
```

التساؤل هنا لماذا وضعنا في الحقل الخاص بال **(button)** هذا الكود :

```
onclick="information()"
```

عزيز المتدرب هنا بما انه حقل خاص بال **(button)** اذا يجب بعد ان يضغط عليه المستخدم ان ينفذ امر معين اذا هنا نضع اسم الدالة اي ال **function** الذي كتبنا بداخله الكود لكي ينفذ الاوامر التي بداخله لهذا وضعنا كلمة **(onclick)** اي مجرد ما يضغط المستخدم على ال **(button)** نفذ الدالة اي ال **(function)** المسمى **(information)** وهو اسم الدالة التي سميناها في اول برنامجنا .. وبمجرد الضغط عليها سوف يقوم البرنامج بعمله وهو بناء جدول وبداخله القيم التي ادخلها المستخدم ..

وهكذا اكون قد حللته لك الى ابعد الحدود وكأنني شرحته وانا ارى انني شرحته فأندمجت في شرحه ولم اشعر بذلك..

اما لماذا لم اكتب شرح بقية الكود فلا يوجد جديد به فكله اوامر **(document.writeln)** اي اوامر طباعه وضع بداخلها كود ال **(html)** اما كيفية كتابة كود ال **(html)** بداخل كود الجافا سكربت ذكرناه في بداية كتابنا .. بقي الكود الذي في ال **(body)** اي في جسم الصفحة هذا عباره عن كود **(html)**..

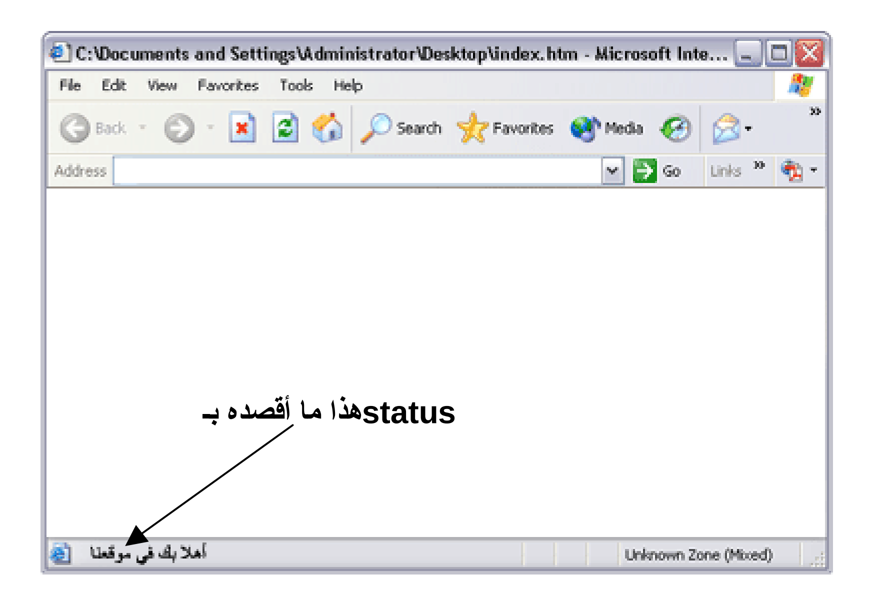
§ اما لماذا هنا كتبنا كود طباعة الجدول في داخل كود الجافا سكربت وايضا كود في الجسم الصفحة ..؟

الجواب :

الاول اي كود انشاء الجدول في الجافا سكربت قصدنا به انه ينشئه في حالة تنفيذ الداله اي بعد ان يضغط المستخدم على ال **(button)** ويحتوي بداخله القيم التي سوف يطبعها ويظهرها للمستخدم اما الجدول الاخر الذي في ال **(body)** هو الجدول الخاص بالنموذج اي ال **(form)** الذي سوف يظهر للمستخدم عند دخوله للصفحة والذي سوف يملأ حقوله لتنتقل الى دالة الجافا سكربت وينفذها ..

ملاحظة :
قد تستخدم شروط معينه تلزم بها معبئ الفورم باستخدام احرف او رموز معينه على سبيل المثال ذكر رمز الـ **(@)** في تعبئة المستخدم خانة البريد الالكتروني للتأكد انه وضع بريد الكتروني وغيرها من الشروط التي أتركها لك لتقوم بها بنفسك لكي لا يكون كل شيء جاهز دائما ولكن سوف ألمح لك كيفية عملها ألا يمكنك عمل داله اي **function** تقوم بفحص خانة معينه مثلا تفحصها بجملة **if** بداخل **function** ويتم استدعاء ال **function** عند تعبئة المستخدم خانة معينه مثل خانة البريد مثلا .

§ والمثال الآخر يتحدث عن طباعة جملة معينة في شريط **(status)** والذي يعتبر من الكائن **window** تمثله هذه الصورة :



فإذا أردت عزيزي المتدرب أن تذكر أي جملة معينة فتظهر في شريط **(status)** تستطيع ذلك من خلال الامر :

```
window.status = " أهلا بك في موقعنا " ;
```

أما إذا كنت تريد أن يظهر في شريط **(status)** متغير كأن يظهر مثلا أسم زائر موقعك فيكتب بهذا الشكل :

```
window.status = name ;
```

ولو أردت أن تذكر في شريط **(status)** جملة ومن ثم متغير فتكون الصيغ كما يلي :

-١

```
window.status = name  " أهلا بك يا " ;
```

.

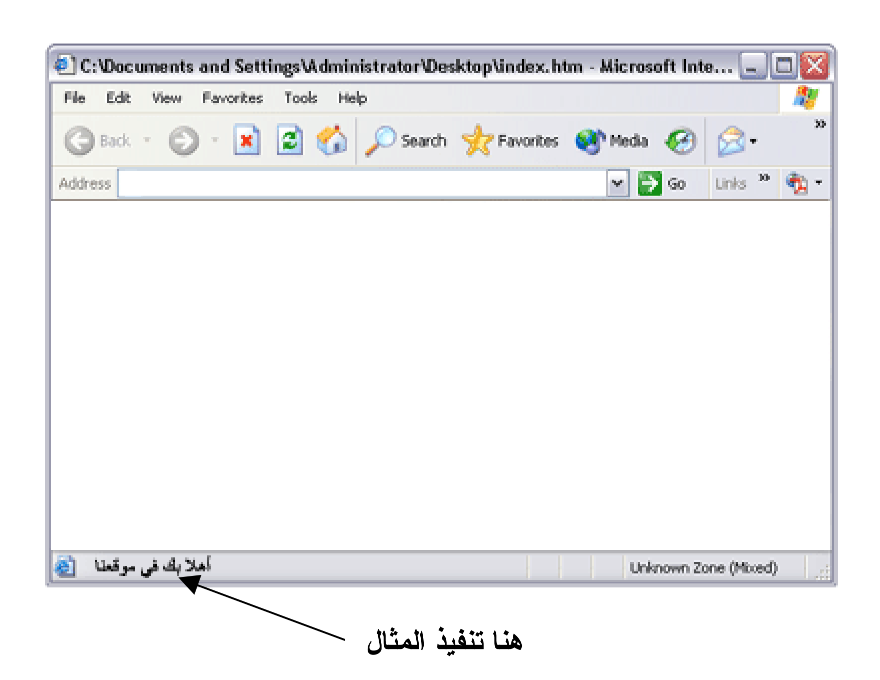
-٢

```
window.status =  " شرفت موقعنا "   name ;
```

مثال :

```
<html >
     <head><title>الجافا سكريبت</title>
           <script type = "text/javascript">
                 window.status = " أهلا بك في موقعنا " ;
           </script>
     </head>
   <body></body>

</html>
```



هنا تنفيذ المثال

لايحتاج الى تحليل فأنت أعلم به . ولكن لنفرض ان لدينا بعض الجمل الموجوده في المصفوفه ونريد عرضها في ال **status** اي يعرض لنا جمل متغيرة باستمرار ... اتركه لك لتفكر به ولكن ألمح لك لبعض الاستخدامات لتنفيذ ذلك.. اولا نحتاج الى تعريف مصفوفة **Array** وايضا الى جملة **for** والبقية عليك .

- لنأخذ ايضا بعض من الكائن **window** كأمثله وسوف اترك لك تحليلها ولكن سوف اضع بعض التعليقات التي سوف تساعدك على تحليل كل مثال ولن اضع لك صورا لها لكي تنفذها انت في نفسك وتتخيلها ..

```
<html >
    <head><title>الجافا سكربت</title>
        <script type = "text/javascript">
          window.open ("www.freewebs.com", "new_web", " toolbar = no ,
          location = no , directories = no , status = no , menubar= no , scrollbars
          = no  , resizable = no , copyhistory = yes , width = 400 , height = 400 " ) ;
        </script>
    </head>
  <body></body>

</html>
```

يقوم هذا البرنامج او الكود بفتح صفحة جديده بمجرد دخول المستخدم على الصفحة التي بها هذا الكود اما ما يحتوي تلاحظ انه يحتوي على الكائن **window** وبداخله اسم الموقع الذي سوف يفتحه للمستخدم اما بقية الاوامر هي خصائص صفحة الاكسبلورر من وجود **toolbar** او عدمه من سماح للمستخدم بالتحكم في حجم الصفحة التي سوف تفتح له بالاضافه الى طول وعرض الصفحة التي تريد ان تفتح للمستخدم ... الخ جرب ان تغير بين ال **yes** وال **no** وانظر ماذا سوف يحدث فالمعلومه تصلك بالتطبيق **..**

```
<html >
    <head><title>الجافا سكربت</title>
        <script type = "text/javascript">
                location.reload( )
        </script>
    </head>
  <body></body>

</html>
```

يقوم الكائن **reload** من خلال الامر **location.reload( )** بإعادة تحميل الصفحة اي عملية **refresh** للصفحة .

ملاحظة:

يفضل إستخدام أمر **break** أو وضع الـ **location.reload( )** ضمن تكنيك معين من دالة أو حلقة تكرار لكي لا يدوم الـ **refresh** الى ما لا نهاية وبالتالي لا تستطيع إغلاق الصفحة

```
<html >
    <head><title>الجافا سكربت</title>
        <script type = "text/javascript">
              window.print( )
        </script>
    </head>
  <body></body>

</html>
```

يقوم هذا الكود بطباعة الصفحة التي يوضع بها الكائن **( )window.print** اي تشغيل خاصية الطباعة في الجهاز .. لتقوم الطابعة بطباعة الصفحة بمعنى بدلا من ان يذهب المستخدم الى **file** ومن ثم يختار **print** هذا الامر يقوم تلقائيا بذلك.. بمجرد دخول الزائر الى الصفحة عزيزي المتدرب حاول ان تضعه انت ضمن **button** بحيث عندما يضغط المستخدم عليها تقوم بطباعة الصفحة.. ليست صعبه تذكر انه يمكننا عمل **function** ووضع بداخله كائن طباعة الصفحة ومن ثم الذهاب الى كود الايقونة في ال **HTML** ووضع **onclick** ومساوته بإسم ال **function** وبهذا نقول عندما يضغط نفذ تذكر ما شرحناه سابقا في الـ **form** .

```
<html  dir="rtl">
    <head><title>الجافا سكربت</title>
        <script type = "text/javascript">
     document.write(" إصدار متصفحك هو: "+ navigator.appName + "<br />") ;
     document.write (window.screen.availWidth + "<br />" ) ;
     document.write( window.screen.availHeight + "<br />") ;
     document.write( window.screen.colorDepth + "<br />") ;
        </script>
    </head>
  <body></body>

</html>
```

هذا المثال لا يحتاج الى تحليل فهو يحتوي على كائنات المتصفح ووضعناها في جمل طباعة لكي يطبع لنا ما سوف تقوم به او بمعنى اصح ما سوف يرجعه لنا كل كائن وسوف اضع بعض الشرح لك لكي يسهل عليك تحليله ..

- **navigator.appName** هذا الكائن سوف يعود لنا بإسم المتصفح لدينا ولاحظ عند وضعه في جملة الطباعة وضعناه كمتغير .
- **window.screen.availWidth** هذا الكائن يقوم بارجاع عرض الشاشه في جهاز المستخدم اي **RESOLUTION SCREEN** .
- **window.screen.availHeight** هذا الكائن يقوم بارجاع طول الشاشه في جهاز المستخدم اي **RESOLUTION SCREEN** .
- **window.screen.colorDepth** هذا الكائن يقوم بارجاع لنا مقدار العمق في الألوان المستخدمه في جهاز المستخدم أليس المستخدم يختار في جهازه من **settings** في خصائص عرض الشاشه ال **color quality** مثلا **32 bit** او **16 bit** ... الخ